Programming: Tryton Unconference, Argonne Looks to Singularity for HPC Code Portability, Mitogen v0.2.4 and More Python Bits

By *Roy Schestowitz*
Created *11/02/2019 - 8:05am*
Submitted by Roy Schestowitz on Monday 11th of February 2019 08:05:49 AM Filed under Development [1]

- **Tryton Unconference 2019: In Marseille on the 6th & 7th of June [2]**

  > We will go in the sunny city of Marseille in south of France on the 6th and 7th of June. Contrary to previous editions of the Tryton Unconferences the coding sprint will be organized during the two days preceding the conference.

- **Argonne Looks to Singularity for HPC Code Portability [3]**

  > Scaling code for massively parallel architectures is a common challenge the scientific community faces. When moving from a system used for development?a personal laptop, for instance, or even a university?s computing cluster?to a large-scale supercomputer like those housed at the Argonne Leadership Computing Facility, researchers traditionally would only migrate the target application: the underlying software stack would be left behind.
  >
  > To help alleviate this problem, the ALCF has deployed the service Singularity. Singularity, an open-source framework originally developed by Lawrence Berkeley National Laboratory (LBNL) and now supported by Sylabs Inc., is a tool for creating and running containers (platforms designed to package code and its dependencies so as to facilitate fast and reliable switching between computing environments)?albeit one intended specifically for scientific workflows and high-performance computing resources.

-

Mitogen v0.2.4 released [4]

> Mitogen for Ansible v0.2.4 has been released. This version is noteworthy as it contains major refinements to the core libary and Ansible extension to improve its behaviour during larger Ansible runs.
>
> Work on scalability is far from complete, as it progresses towards inclusion of a patch held back since last summer to introduce per-CPU multiplexers. The current idea is to exhaust profiling gains from a single process before landing it, as all single-CPU gains continue to apply in that case, and there is much less risk of inefficiency being hidden in noise created by multiple multiplexer processes.

- **Introducing kids to computational thinking with Python** [5]
- **The Factory Method Pattern and Its Implementation in Python** [6]
- **PyDev of the Week: Paolo Melchiorre** [7]
- **Create a filter for the audio and image files with python** [8]
- **Some simple CodeWars problems** [9]

Development

Source URL: http://www.tuxmachines.org/node/120542

Links:
[1] http://www.tuxmachines.org/taxonomy/term/145
[2] https://discuss.tryton.org/t/tryton-unconference-2019-in-marseille-on-the-6th-7th-of-june/1125
[3] https://insidehpc.com/2019/02/argonne-looks-to-singularity-for-hpc-code-portability/
[4] https://sweetness.hmmz.org/2019-02-10-mitogen-v0-2-4.html
[5] https://opensource.com/article/19/2/break-down-stereotypes-python
[6] https://realpython.com/factory-method-python/
[7] http://www.blog.pythonlibrary.org/2019/02/11/pydev-of-the-week-paolo-melchiorre/
[8] http://codingdirectional.info/2019/02/11/create-a-filter-for-the-audio-and-image-files-with-python/
[9] https://www.codementor.io/mikebell66/some-simple-codewars-problems-s2gf1tkne