ML and GPU Accelerated Visualization with JupyterHub

Markus Blank-Burian[1] Jürgen Hölters[1] Raimund Vogl[1]

WWU Münster, Germany
{blankburian,holters,rvogl}@uni-muenster.de

Abstract

At the IT department of WWU Münster (WWU IT), we operate a JupyterHub for education and research. Researchers have access to virtual GPUs from their Jupyter sessions. These may be used natively to compile and run CUDA accelerated code, e.g. for machine learning. Using VirtualGL, we also provide an accelerated X server in Jupyter sessions. X11 applications are then accessible from the browser using noVNC.

1 Introduction

Jupyter notebooks [2, 4] are a defacto standard for scientific research. They provide an easy to use interface for visualizing results in a reproducable manner. Lately, the traditional notebook interface has been replaced by JupyterLab. JupyterLab provides a nice tree view for file selection, a tabbed interface for opening multiple notebooks and a command line terminal. From a launcher page, notebooks and other applications may be started.

At the IT department of WWU Münster (WWU IT), we operate a JupyterHub using KubeSpawner on a Kubernetes cluster [1]. This Kubernetes cluster itself runs on WWU Cloud, a private IaaS cloud based on OpenStack [5] and part of the RDI-NRW project [6]. The JupyterHub can be used by students, in teaching and by researchers. We provide a selection of notebook images containing standard applications available in the JupyterLab user interface. Sessions in JupyterHub may have access to virtualized GPUs, used by CUDA applications and for visualization.

2 WWU Cloud

The WWU Cloud is based on OpenStack and Ceph, providing IaaS services for WWU IT and researchers. Once a project in OpenStack is created, virtualized infrastructure can be created on demand. To orchestrate the OpenStack services, we use a baremetal Kubernetes cluster.

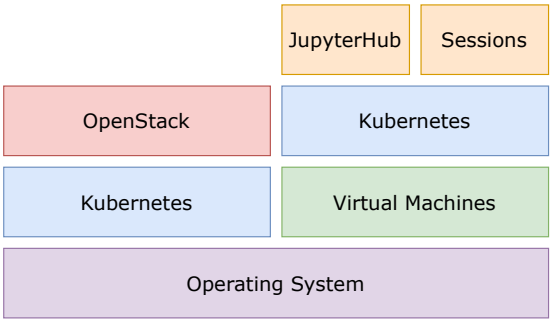


Figure 1: Software Stack

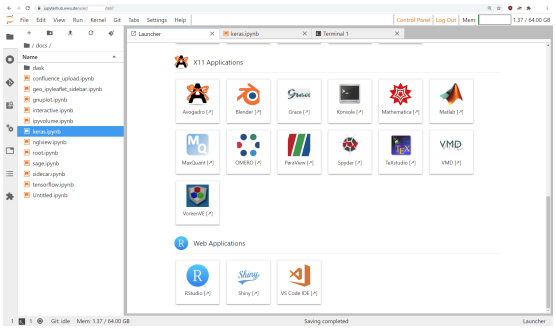


Figure 2: JupyterLab Launcher

Using NVidia GRID technology, GPUs are virtualized as mediated devices "mdev". Device type selection in Nova is currently limited to a single type per card. In the future, this limitation can be avoided with OpenStack Cyborg, making it possible to dynamically select an arbitrary vGPU type for a VM.

All types of virtual GPUs are available in OpenStack as custom resources named "VGPU", making distinguishing between GPU types difficult. As we have only a single GPU type per node, we added a special node trait on every node, specifying the GPU type. This makes it possible to define flavors with vGPUs, allowing a VM to select the GPU type.

On OpenStack, WWU IT operates a kubernetes cluster, which is used for the deployment of JupyterHub. Kubernetes nodes are organized in multiple groups: control plane, nodes, worker nodes and worker nodes with vGPU. Control plane and node VMs have anti-affinity and are suited for all types of services, whereas workers are used for JupyterHub Sessions. Nodes with vGPUs are automatically labeled using "node-feature-discovery" and the "nvidia-device-plugin". Pods can select the GPU type directly via node label.

3 JupyterHub

JupyterHub is a webservice, providing jupyter notebook sessions hosted on dedicated hardware. WWU IT operates a JupyterHub deployment on its kubernetes cluster, available for all students and employees of WWU, mostly used by researchers and for courses. JupyterHub is configured to use "KubeSpawner" to spawn new instances as Pods. After login, users are presented with a dialog for selecting the cpu and ram limits as well as the notebook image and vGPU support. WWU IT provides a small selection of notebook images, containing commonly used software for data analysis and visualization.

JupyterLab is used as the main user interface, providing a launcher interface to start applications, notebooks and terminal sessions. Combined with a treeview for file selection, git integration and a tabbed interface for opening multiple notebooks, JupyterLab provides a complete virtualized working environment in the browser. Our file storage in WWU Cloud as well as our HPC cluster filesystems are directly available from JupyterLab sessions, making it ideal to quickly analyze and visualize results from computations.

Apart from notebooks, we offer starting browser based applications like "code-server" (a VS-Code fork) and R Studio. This is realized using websockets and port-forwarding in JupyterLab. These web applications run natively in the browser and are therefore very responsive.

We also offer X11 applications in JupyterHub sessions, realized by using noVNC. noVNC

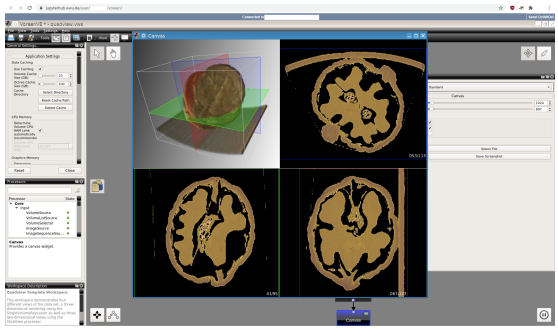


Figure 3: Voreen [3], an X11 3D visualization application, started inside a JupyterHub session

is a browser based remote visualization software based on the VNC protocol. The applications are rendered on the Xorg server in the JupyterHub sessions and the output is compressed and transfered to the browser, where it is displayed. In case, the session is started with vGPU support, we configure an Xorg server with VirtualGL in a sidecar container, so that multiple applications in a single session share a single GPU accelerated framebuffer. X11 Applications see the full set of GLX extensions provided by the native driver and use server based rendering. At WWU IT, we can therefore offer high end GPU acceleration for data visualization, with no additional requirements on the client side.

When vGPU support is selected, the virtual GPUs are also directly accessible from within the JupyterHub sessions. CUDA applications can be compiled and executed natively, e.g. for machine learning tasks. We also integrated major machine learning frameworks (TensorFlow, Kiras, PyTorch) into our notebook images.

4 Acknowledgements

- We thank MKW NRW for funding the RDI-NRW project.

References

[1] Brendan Burns, Joe Beda, and Kelsey Hightower. *Kubernetes: up and running: dive into the future of infrastructure*. O'Reilly Media, 2019.

[2] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. *Jupyter Notebooks-a publishing format for reproducible computational workflows.*, volume 2016. 2016.

[3] Jennis Meyer-Spradow, Timo Ropinski, Jörg Mensmann, and Klaus Hinrichs. Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. *IEEE Computer Graphics and Applications*, 29(6):6–13, 2009.

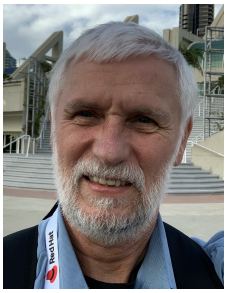
[4] Jeffrey M Perkel. Why jupyter is data scientists' computational notebook of choice. *Nature*, 563(7732):145–147, 2018.

[5] Tiago Rosado and Jorge Bernardino. An overview of openstack architecture. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, pages 366–367, 2014.

[6] Raimund Vogl, Dominik Rudolph, and Anne Thoring. Bringing structure to research data management through a pervasive, scalable and sustainable research data infrastructure. In *The Art of Structuring*, pages 501–512. Springer, 2019.

5 Biographies

M. Blank-Burian is a research assistant at the it department of the University of Münster (WWU IT) and lead cloud architect of the WWU Cloud. He graduated from University of Münster (Germany) in 2013 with degrees in both physics and computer sciences. In 2018, he received his Ph.D. in theoretical physics. His research focuses on private clouds and cloud native technology. More info: https://www.uni-muenster.de/forschungaz/person/17330



J. Hölters is the head of the department for systems at it department of the University of Münster (WWU IT). More info: https://www.uni-muenster.de/forschungaz/person/8172

R. Vogl holds a Ph.D. in elementary particle physics from the University of Innsbruck (Austria). After completing his Ph.D. studies in 1995, he joined Innsbruck University Hospital as IT manager for medical image data solutions and moved on to be deputy head of IT. He served as a lecturer in medical informatics at UMIT (Hall, Austria) and as managing director for a medical image data management software company (icoserve, Innsbruck) and for a center of excellence in medical informatics (HITT, Innsbruck). Since 2007 he has been director of the it department of the University of Münster (WWU IT, Germany). His research interests focus on management of complex information systems and information infrastructures.