DGX-1

<table>
<thead>
<tr>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
</tr>
<tr>
<td>CPU Cores per node</td>
</tr>
<tr>
<td>CUDA Cores</td>
</tr>
<tr>
<td>Memory (DDR4) per node</td>
</tr>
<tr>
<td>Peak Performance</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software (OS and development environment)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
</tr>
<tr>
<td>MPI</td>
</tr>
<tr>
<td>Compilers</td>
</tr>
<tr>
<td>Performance libraries</td>
</tr>
<tr>
<td>Tools for performance and correctness analysis</td>
</tr>
</tbody>
</table>

User Guide
DGX-1-UserGuide.pdf

Machine Learning System Nvidia DGX-1 and OpenStack GPU VMs

Intro

The Machine Learning System DGX-1 is a “Supercomputer in a box” with a peak performance of 170 TFlop/s (FP16). It contains eight high end GPGPUs from NVIDIA (Tesla P100) with each 16 GB RAM and 28.672 CUDA-compute units which are connected to each other by a NVLink Interconnect and a host x86 compatible system with 40 cores (Intel Xeon). Users can reserve the whole DGX-1 exclusively and run complex machine-learning tasks, which are available via Docker images.

A set of preinstalled images covers deep learning toolkits such as TensorFlow, Theano, CNTK, Torch, DIGITS, Caffe and others.

The system is running Ubuntu 14.04 LTS in the version supported by NVidia.

Below is a scematic drawing about the internals of the system. The 8 GPUs are connected via NVlink High Speed Interconnect and the x86 cores are connected via PCIe Switches to the GPUs. For a detailed documentation about the hardware please see the Nvidia website directly.

Also available are 4 single node systems with one NVIDIA GPU P100 for development purposes. On these systems only a general purpose image is available which provides the PGI Compiler Suite and CUDA.

**NVIDIA® NVLink™ Hybrid Cube Mesh**
Access and Login

**LRZ Linux Cluster Users** (Here is how can you get an LRZ Linux Cluster account: [https://www.lrz.de/services/compute/application/](https://www.lrz.de/services/compute/application/), please note: your tum /lmu or Irz account is not an LRZ Linux Cluster account!!!) can get access to the system by submitting an Incident ticket (subject linux cluster) to the LRZ service desk (servicedesk@lrz.de). Please briefly describe your research in your application and notice that DGX-1 is in high demand, and it is a production machine, meaning it should not be used as a debugging tool. Any software you want to run on DGX-1 should have been already tested with GPU elsewhere. Currently all of LRZ GPU systems are in testing phase, there is no backup for data. You have to backup your data by yourself.

The system can be reserved via an online calendar system ([https://datalab.srv.lrz.de](https://datalab.srv.lrz.de)) which is at the moment only available in the MWN. If you want to use the reservation system and the login to the compute system from outside the MWN you first have to connect to the LRZ VPN (see VPN documentation).

In the online calendar reservation the user can see the available timeslots and book the complete system for maximal 6 hours per slot.

Please remember that on DGX-1 only /home/ is kept between sessions. On VMs **everything** is lost. We are working on that but in the meantime always copy your data.

The user has then to upload a ssh key into the online calendar which will be used for authentication on the system.

When the date of the reservation approaches, the user will obtain an email with further instructions how to connect to the system via ssh and via a http link. Please be aware that the users computer has to be in the Munich Science Network or LRZ VPN in order to connect to the system.

NVIDIA GPU Optimized Deep Learning Frameworks

The NVIDIA Deep Learning SDK accelerates widely-used deep learning frameworks.
This release provides containerized versions of those frameworks optimized for the **NVIDIA DGX-1**, pre-built, tested, and ready to run, including all necessary dependencies. Together with [Nvidia deep learning institute](http://www.nvidia.com), LRZ regularly offers Deep Learning Workshops on site, please check [LRZ workshop course page](http://www.lrz.de/services/courses/).

<table>
<thead>
<tr>
<th>Framework</th>
<th>Base Version</th>
<th>Container Name</th>
<th>Description</th>
<th>Release Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe</td>
<td>0.16.4</td>
<td>Caffe</td>
<td>Caffe was originally developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors. It is a deep learning framework made with expression, speed and modularity in mind. NVIDIA Caffe is an NVIDIA-maintained fork of BVLC Caffe tuned for NVIDIA GPUs, particularly in multi-GPU configurations, accelerated by the NVIDIA Deep Learning SDK. It includes multi-precision support as well as other NVIDIA-enhanced features and offers performance specially tuned for the NVIDIA DGX-1. NVIDIA's Manual for this version is available <a href="http://www.nvidia.com">here</a>.</td>
<td>Link</td>
</tr>
<tr>
<td>Caffe2</td>
<td>0.8.1</td>
<td>Caffe2</td>
<td>Caffe2 is a deep-learning framework designed to easily express all model types, for example, CNN, RNN, and more, in a friendly python-based API, and execute them using a highly efficiently C++ and CUDA back-end. It allows a large amount of flexibility for the user to assemble their model, whether for inference or training, using combinations of high-level and expressive operations, before running through the same python interface allowing for easy visualization, or serializing the created model and directly using the underlying C++ implementation. Caffe2 supports single and multi-GPU execution, along with support for multi-node execution.</td>
<td>Link</td>
</tr>
<tr>
<td>CNTK</td>
<td>2.2</td>
<td>CNTK</td>
<td>Microsoft Cognitive Toolkit (CNTK) empowers you to harness the intelligence within massive datasets through deep learning by providing uncompromised scaling, speed and accuracy with commercial-grade quality and compatibility with the programming languages and algorithms you already use.</td>
<td>Link</td>
</tr>
<tr>
<td>DIGITS</td>
<td>0.11.0</td>
<td>DIGITS</td>
<td>DIGITS can be used to rapidly train highly accurate deep neural network (DNNs) for image classification, segmentation and object detection tasks. DIGITS simplifies common deep learning tasks such as managing data, designing and training neural networks on multi-GPU systems, monitoring performance in real time with advanced visualizations, and selecting the best performing model from the results browser for deployment. DIGITS is completely interactive so that data scientists can focus on designing and training networks rather than programming and debugging.</td>
<td>Link</td>
</tr>
<tr>
<td>MXNet</td>
<td>0.16.4</td>
<td>MXNet</td>
<td>MXNet is a deep learning framework designed for both efficiency and flexibility. It allows you to mix the flavors of symbolic programming and imperative programming to maximize efficiency and productivity. In its core is a dynamic dependency scheduler that automatically parallelizes both symbolic and imperative operations on the fly. A graph optimization layer on top that makes symbolic execution fast and memory efficient. The library is portable and lightweight, and it scales to multiple GPUs and multiple machines. MXNet is also more than a deep learning project. It is also a collection of blueprints and guidelines for building deep learning systems and interesting insights of DL systems for hackers.</td>
<td>Link</td>
</tr>
</tbody>
</table>
| PyTorch    | v0.2         | PyTorch        | PyTorch is a python package that provides two high-level features:  
  - Tensor computation (like numpy) with strong GPU acceleration  
  - Deep Neural Networks built on a tape-based autograd system  
You can reuse your favorite python packages such as numpy, scipy and Cython to extend PyTorch when needed. | Link          |
| TensorFlow | 1.3.0        | TensorFlow     | TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture lets you deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device without rewriting code. TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well. | Link          |
| Theano     | 0.10         | Theano         | Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. | Link          |
Torch is a scientific computing framework with wide support for deep learning algorithms. Thanks to an easy and fast scripting language, Lua, and an underlying C/CUDA implementation, Torch is easy to use and is efficient.

Torch offers popular neural network and optimization libraries that are easy to use yet provide maximum flexibility to build complex neural network topologies.

General purpose container

CUDA and OpenACC compiler ("CUDA 9 and PGI 17.10")

This is a basic container for software development containing the following components:

- Ubuntu 16.04
- NVIDIA CUDA® 9.0.176
- NVIDIA cuDNN 7.0.3
- NVIDIA NCCL 2.0.5 (optimized for NVLink)
- PGI C++ and Fortran Compiler 17.4
- OpenMPI with GPUDirect

Using nvidia-docker on single GPU virtual servers

As single GPU virtual servers are deleted completely after each session a quick way to install your software is helpful. The fastest way to run a machine learning application on the single GPU system is to use nvidia-docker:

Example Tensorflow:
Log in to virtual server using

```
ssh -L 8888:localhost:8888 ubuntu@<your ip address from setup email>
```

Pull and start the container:

```
nvidia-docker run -it -p 8888:8888 gcr.io/tensorflow/tensorflow:latest-gpu
```

Then open http://localhost:8888 in your browser.
You can use the option "-v /ssdtemp:/ssdtemp" to map the 800 GB of storage space mounten on /ssdtemp in the server to the container.
A list of Tensorflow containers is available here: https://hub.docker.com/r/tensorflow/tensorflow/tags/
You can of course package your own applications using nvidia-docker, see https://devblogs.nvidia.com/parallelforall/nvidia-docker-gpu-server-application-deployment-made-easy/

Changelog

11. January 2018

Single GPUs updated to CUDA 9.1 and PGI 17.9

2. November 2017

Updated all containers to 17.10 - old containers for MXnet, CUDA, Tensorflow and Caffe are still available. 17.10 is basically only a minor update to CUDA, cuDNN and NCCL and includes latest OS updates.