

Installation Tensorflow mit GPU-Nutzung für OpenSuse 15.0 oder 15.1

CUDA sollte schon vorinstalliert sein!

Alle Befehle bitte als root-User ausführen.

Python

Es wird python 3.6 verwendet. Version überprüfen:

- `python --version`

Version kann wie folgt angepasst werden:

- `update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1`
- `update-alternatives --install /usr/bin/python python /usr/bin/python3.6 2`

check Version:

- `python --version`

list versions:

- `update-alternatives --list python`

change version:

- `update-alternatives --list python`
selection number: 2

Folgende Pakete sollten vorerst über YAST2 installiert werden:

- `python-devel`
- `python3-kafka-python`
- `patch`
- `python3-six`
- `python3-numpyexpr`
- `python3-numpy`
- `python3-numpy-devel`
- `python-numpydoc`
- `python3-wheel`
- `python3-mock`
- `python3-setuptools`

Pakete können auch über die Kommandozeile mit pip install installiert werden:

- `pip install pip six numpy wheel setuptools mock`

Mit pip install muss noch folgendes installiert werden:

- `pip install future>=0.17.1`
- `pip install keras_applications==1.0.6 --no-deps`
- `pip install keras_preprocessing==1.0.5 --no-deps`
- `pip install h5py`

Weitere Pakete installieren

- zypper install patch
- zypper install python-devel
- zypper install apache-kafka

Bazel

Download Version von hier: <https://github.com/bazelbuild/bazel/releases>

Für Tensorflow 1.14 bitte Bazel Version 24.1 oder höher verwenden. Ggf. muss die Version nochmal angepasst werden.

Installation:

- `chmod +x bazel-<version>-installer-linux-x86_64.sh`
- `./bazel-<version>-installer-linux-x86_64.sh`

NCCL (falls benötigt)

Download tar-datei von <https://developer.nvidia.com/developer-program>

! Account wird benötigt !

Heruntergeladene tar-datei nach /usr/local verschieben.

Datei entpacken (Version beachten):

- `tar -xvf nccl_2.4.7-1+cuda10.1_x86_64.txz`

Pfad zur .bashrc-Datei hinzufügen:

- `vim ~/.bashrc`

Folgenden Pfad in letzte Zeile hinzufügen (Version beachten):

- `/usr/local/nccl_2.4.7-1+cuda10.1_x86_64`

Tensorflow

Installation erfolgt mit git. Falls benötigt git nachinstallieren:

- `zypper in git`

Zuerst aber Verzeichnis wechseln:

- `cd /usr/local/src`

Nun kann das Verzeichnis geklont bzw. heruntergeladen werden:

- `git clone https://github.com/tensorflow/tensorflow.git`

Ins tensorflow-Verzeichnis wechseln

- `cd tensorflow`

Bei Bedarf kann die Tensorflow Version einfach geändert werden (über git wird immer die aktuellste Version heruntergeladen):

- `git checkout <version>` z. B. r1.12, r1.13, r1.14

Nun kann tensorflow konfiguriert werden:

- ./configure

Tensorflow kann zum Beispiel wie folgt konfiguriert werden:

(Die Konfiguration kann man jederzeit unterbrechen oder von neuem beginnen)

```
WARNING: --batch mode is deprecated. Please instead explicitly shut down your
Bazel server using the command "bazel shutdown".
INFO: Invocation ID: 49088008-0dbf-4b93-ade6-8fd6b0650969
You have bazel 0.22.0 installed.
Please specify the location of python. [Default is /usr/bin/python]:
Found possible Python library paths:
/usr/lib/python3.6/site-packages
/usr/lib64/python3.6/site-packages
Please input the desired Python library path to use. Default is
[/usr/lib/python3.6/site-packages]
Do you wish to build TensorFlow with XLA JIT support? [Y/n]: Y
XLA JIT support will be enabled for TensorFlow.
Do you wish to build TensorFlow with OpenCL SYCL support? [y/N]: N
No OpenCL SYCL support will be enabled for TensorFlow.
Do you wish to build TensorFlow with ROCm support? [y/N]: N
No ROCm support will be enabled for TensorFlow.
Do you wish to build TensorFlow with CUDA support? [y/N]: y
CUDA support will be enabled for TensorFlow.
Projektarbeit Sommer 2019, Nathanaël Taton, Physikalisches Institut Heidelberg
Seite | 25Please specify the CUDA SDK version you want to use. [Leave empty to default
to CUDA 10.0]:
Please specify the location where CUDA 10.0 toolkit is installed. Refer to
README.md for more details. [Default is /usr/local/cuda]:
Please specify the cuDNN version you want to use. [Leave empty to default to
cuDNN 7]:
Please specify the location where cuDNN 7 library is installed. Refer to
README.md for more details. [Default is /usr/local/cuda]:
Do you wish to build TensorFlow with TensorRT support? [y/N]: N
No TensorRT support will be enabled for TensorFlow.
Please specify the locally installed NCCL version you want to use. [Default is to
use https://github.com/nvidia/nccl]:
Please specify a list of comma-separated Cuda compute capabilities you want to
build with.
You can find the compute capability of your device at:
https://developer.nvidia.com/cuda-gpus.
Please note that each additional compute capability significantly increases your
build time and binary size. [Default is: 5.2,6.1]: 6.1
Do you want to use clang as CUDA compiler? [y/N]: N
nvcc will be used as CUDA compiler.
Please specify which gcc should be used by nvcc as the host compiler. [Default is
/usr/bin/gcc]:
Do you wish to build TensorFlow with MPI support? [y/N]: N
No MPI support will be enabled for TensorFlow.
Please specify optimization flags to use during compilation when bazel option "--
config=opt" is specified [Default is -march=native -Wno-sign-compare]:
Would you like to interactively configure ./WORKSPACE for Android builds? [y/N]:
```

N

Not configuring the WORKSPACE for Android builds.

Preconfigured Bazel build configs. You can use any of the below by adding "--config=<>" to your build command. See .bazelrc for more details.

--config=mkl

--config=monolithic

--config=gdr

Build with MKL support.

Config for mostly static monolithic build.

Build with GDR support.

--config=verbs # Build with libverbs support.

--config=ngraph # Build with Intel nGraph support.

--config=dynamic_kernels # (Experimental) Build kernels into separate shared objects.

Preconfigured Bazel build configs to DISABLE default on features:

--config=noaws # Disable AWS S3 filesystem support.

--config=nogcp # Disable GCP support.

--config=nohdfs # Disable HDFS support.

--config=noignite # Disable Apache Ignite support.

--config=nokafka # Disable Apache Kafka support.

--config=nonccl # Disable NVIDIA NCCL support.

Configuration finished

Als nächstes wird tensorflow kompiliert werden:

- `bazel build --config=opt --config=cuda --cxxopt="-D_GLIBCXX_USE_CXX11_ABI=0" //tensorflow/tools/pip_package:build_pip_package`

Jetzt noch das Paket erstellen und installieren:

- `./bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg`
- `pip install /tmp/tensorflow_pkg/tensorflow-1.14.0-cp36-cp36m-linux_x86_64.whl`
(Tensorflow-Version kann ggf. eine andere sein)

Tensorflow Version überprüfen:

- `pip show tensorflow`
- Beispiel Ausgabe:
`Name: tensorflow`
`Version: 1.14.0`
`Summary: TensorFlow is an open source machine learning framework for everyone.`
`Home-page: https://www.tensorflow.org/`
`Author: Google Inc.`
`Author-email: packages@tensorflow.org`
`License: Apache 2.0`
`Location: /usr/lib64/python3.6/site-packages`
`Requires: six, wheel, numpy, gast, tensorboard, absl-py, google-pasta, grpcio, keras-applications, protobuf, tensorflow-estimator, termcolor, keras-preprocessing, astor`

Tensorflow sollte nun erfolgreich installiert sein!