**Table of Contents**

[1. About this document 2](#_Toc440915125)

[2. Dependencies 2](#_Toc440915126)

[2.1. CMake 2](#_Toc440915127)

[2.2. MPI (Message Passing Interface) Implementation 2](#_Toc440915128)

[2.2.1. Open MPI 2](#_Toc440915129)

[2.2.2. MVAPICH2 2](#_Toc440915130)

[2.3. CUDA Toolkit 2](#_Toc440915131)

[3. Installation 3](#_Toc440915132)

[3.1. Configure environment and create workspace 3](#_Toc440915133)

[3.2. Build the source 5](#_Toc440915134)

[3.2.1. Build the library 5](#_Toc440915135)

[3.2.2. Build samples 5](#_Toc440915136)

[3.3. Install the library 5](#_Toc440915137)

# 1. About this document

This document describes necessary steps to install the gpusort library correctly.

# 2. Dependencies

This section list all tools and libraries which are required to compile and install the gpusort library from source successfully.

## 2.1. CMake

We use CMake to control the library compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment.

Required version: **2.8 or higher**.

Homepage: [*https://cmake.org/*](https://cmake.org/)

## 2.2. MPI (Message Passing Interface) Implementation

The gpusort library needs a MPI implementation library to be built and run jobs among multiple processes (nodes). One can install one of the following libraries due to the gpusort is tested well with them:

### 2.2.1. Open MPI

Required version: **1.8.5 or higher**.

Homepage: [*http://www.open-mpi.org/*](http://www.open-mpi.org/)

### 2.2.2. MVAPICH2

Required version: **2.2a or higher**.

Homepage: [*http://mvapich.cse.ohio-state.edu/*](http://mvapich.cse.ohio-state.edu/)

## 2.3. CUDA Toolkit

The gpusort library includes some snippets of CUDA code to be able to use Thrust library to sort data using GPU devices. Therefore a CUDA Toolkit is required.

Required version: **6.5 or higher**.

Homepage: [*https://developer.nvidia.com/cuda-toolkit-archive*](https://developer.nvidia.com/cuda-toolkit-archive)

# 3. Installation

## 3.1. Configure environment and create workspace

* Create workspace home directory:



* Configure environment variables and generate makefiles:



with:

+ MPI\_ROOT\_DIR: an absolute path to the MPI library’s installed directory.

+ CUDA\_TOOLKIT\_ROOT\_DIR: an absolute path to the CUDA toolkit’s installed directory.

+ CUDA\_ARCH\_ID: specify the short name of the class of NVIDIA virtual GPU architecture for which the CUDA input files must be compiled. Default value is 20, which is equivalent to these options –arch=compute\_20 and –code=sm\_20 are passed to the CUDA compiler.

+ DO\_BENCHMARK: if this option is set, the gpusort library will show benchmark information after finish sorting data.

+ DEBUG\_MSG: if this option is set, the gpusort library will show debug messages while sorting data.

+ USE\_CPP\_11: if this option is set, the MPI compiler will use C++11 to build the gpusort library.

+ USE\_CPP\_0X: if this option is set, the MPI compiler will use C++0x to build the gpusort library.

+ CMAKE\_INSTALL\_PREFIX: if this variable is set, the installation target directory is changed to this path. Default installation path is /usr/local/gpusort.

For example, please use the following commands to create a workspace home directory at *~/build\_gpusort* and configure environment variables and generate makefiles, so that the target CUDA devices is Tesla K80s, the MPI compiler uses C++0X, the target library will show benchmark information after finish sorting data, and the installation path is changed to *~/install\_gpusort*:



## 3.2. Build the source

### 3.2.1. Build the library



### 3.2.2. Build samples

Please use the following command to compile all example code in the released folder:



## 3.3. Install the library

- In case of using the default installation path:



- In case of using a customized installation path (doesn’t need a root priviledge):

