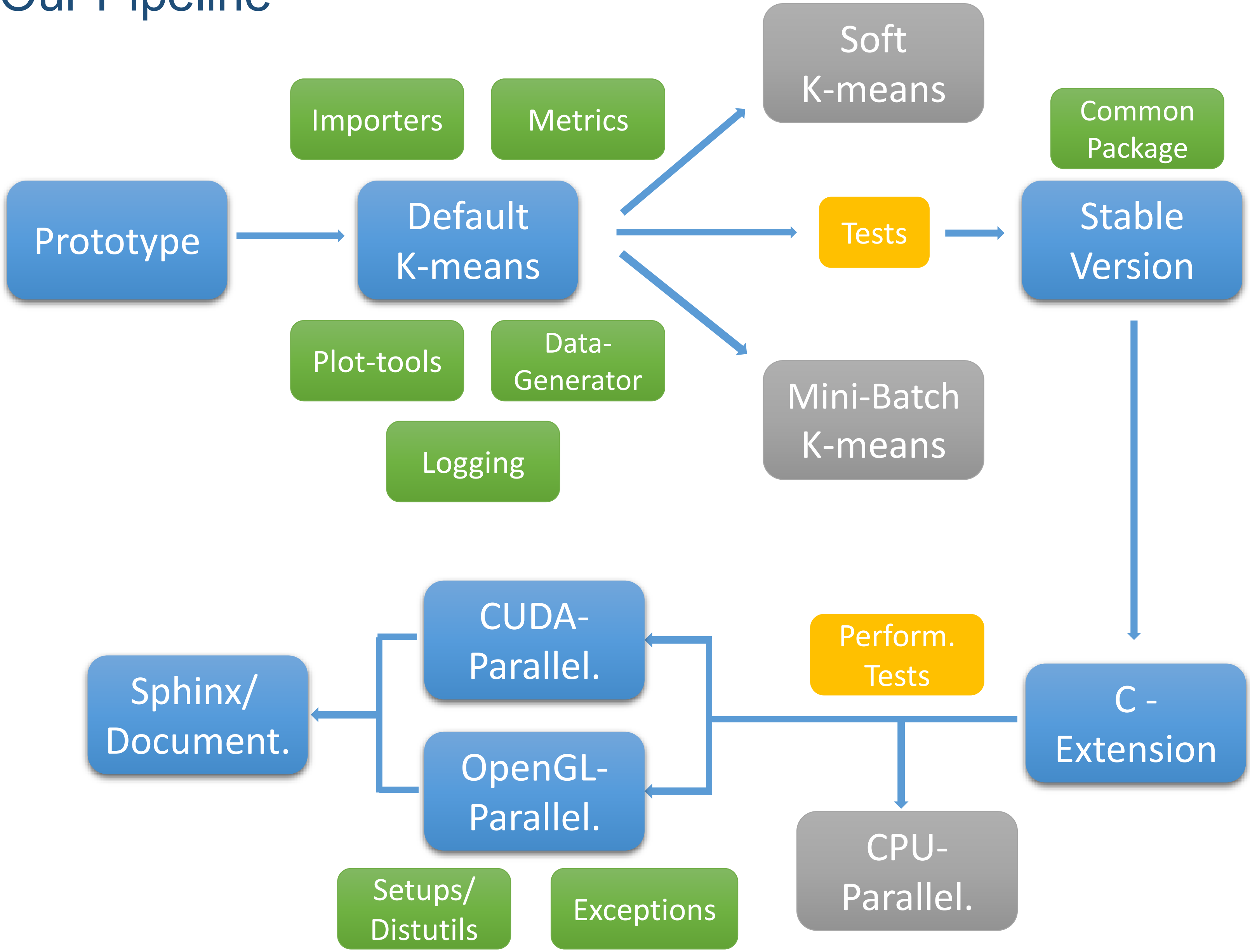
Software Project: Contributions

February 6, 2014

Our Pipeline



GPU Implementations

CUDA	vs.	OpenCL
Inherits C extension		Inherits default python implementation
Executes center assignment on the GPU		Executes whole iteration on the GPU
Supported on NVIDIA cards		Supported on AMD and Intel cards
Speed up of 5-20x		(Likely) a little slower
CUDA environment setups		Usage of PyOpenCL, pip support

Contributions: Moritz

- toy model
- (most of) k-means unittests
- kmeans_main for selecting the best implementation
- architecture of default k-means implementation and data importer
- OpenCL implementation
- logging
- some setup.py files
- simple and text file data importer

Contributions: Florian

- General: Set up git and mailing list, some email support
- k-means: helper classes, such as KmeansPlot and KmeansRandomDataGenerator
- parts of the default python implementation of k-means
- tested mini batch version of k-means
- simple benchmarking using timeit, profiling of the code using line_profiler
- added BinaryFileDataImporter for importing data efficiently using memmaps
- created common package with data importers
- documentation of the code, setup of sphinx for auto-generated html documentation

Contributions: Yuchun

- toy model, not so pythonic, thus abandoned.
- soft k-means implementation.
- fixed some bugs in data importer & default kmeans.
- C-extension implementation, tests.
- CUDA implementation, tests, optimization.
- performance tests among default, C-&CUDA k-means.
- exception handlings for the built-in modules.
- documentation of C-extension & CUDA codes in files & Sphinx.
- rewrite the build functions in distutils package to support nvcc &*.cu files.