Features needed:

1. Create an installer and Linux porting.
2. Make a good error system. Search for all asserts and switch to logs, output them in a log file.
3. Send the program and results via email.
4. Visual tool for defining modules, evaluating potential parallelism etc.
5. Create a better ModulesAnalyzer tool. Include the C/C++ grammar in the parser such that we don’t have restrictions for how the user writes code.
6. Automatic testing of all examples with clear input/output.
7. Garbage collector.

Known bugs:

1. Memory leak in void Master::OnTaskCreated(void\* pTaskData, int iTaskSize), and double copies for BufferTypes
2. Report errors when trying to compose WHILE/FOR modules on the iterating side with modules which are not atomic – we can’t know how the input will flow because a vector of processes interface can have an unknown number of elements.

Documentation:

1. Code structure and classes, the rest is in the Phd thesis..