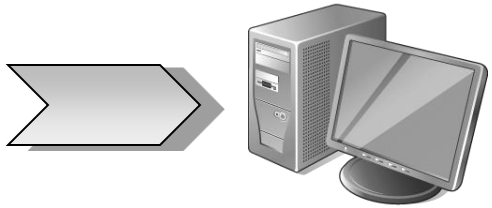
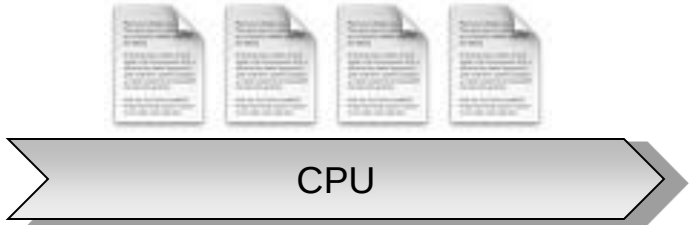
ROSEdu
summer of code

WHC IDE

Workflow of Heterogeneous Computing

Andrei Preda
Veaceslav Munteanu

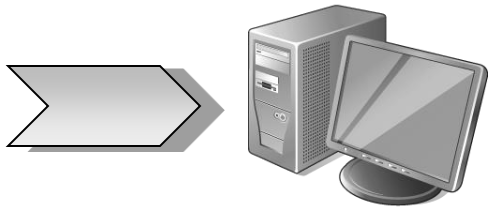


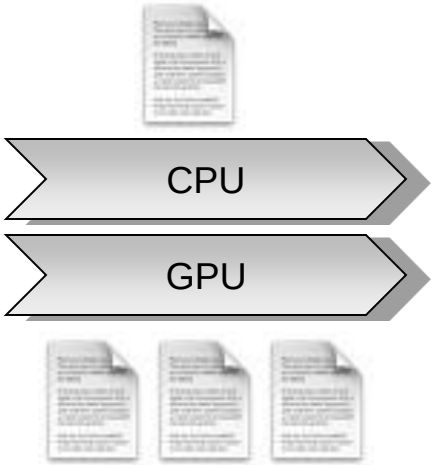


Execution Time

OpenCL

Framework for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs





Execution Time

General info (1)

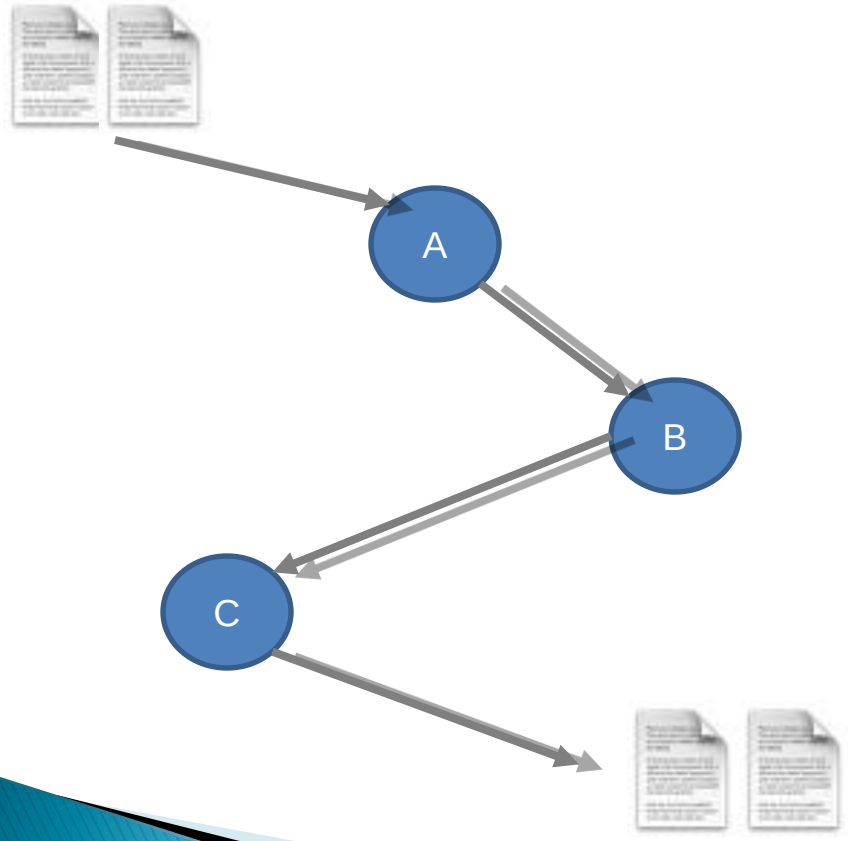
- WHC IDE is an OpenCL IDE
- Developed by UPB students
- Uses the Qt framework

- Simple editor (in development)
- Execution restore in case of crash
- Logging and stats
- Workflow diagram

General info (2)

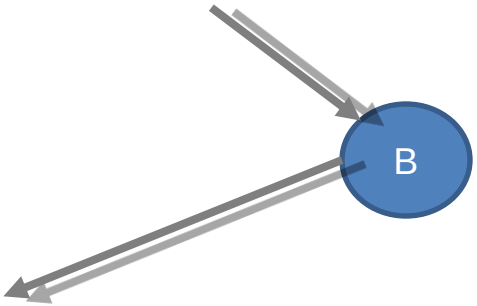
Define Workflow

Input Files



Output Files

Define Task Source Code



BBB.cpp
BBB.cl
BBB.hpp

BBB.cpp

```
// Task generated code
// Respect argument schema
// TASK -in [...] -out [...] -args [optional]

#define IN 1
#define OUT 1

int main(int argc,char** argv)
{
        if(argc<3+IN+OUT)
                return 1;
        else if(argv[1][0]!='-')
                return 1;
        else if(argv[2+IN][0]!='-')
                return 1;
        else if(argv[3+IN+OUT][0]!='-')
                return 1;

// .. TODO add code here
        return 0;
}
```

RSoC goals

- WHC IDE is an OpenCL IDE
- Port to Qt 5
- Fix crashes
- Find memory leaks
- Create a restoring system (checkpointing)
- Add logging and statistics
- Increase execution workflow speed
- Improve editor

Porting to Qt5

- The first step of the project
- Classes are split in modules
- Other modules compared to Qt4
- Some methods changed from Qt4
- The hardest part was Cmake

- Second part of the project
- Gave me a more stable project

Checkpointing

- Refactored execution code
- Made it use the QProcess signals
- Exit code and additional information saved
- The save file can be used to restore execution

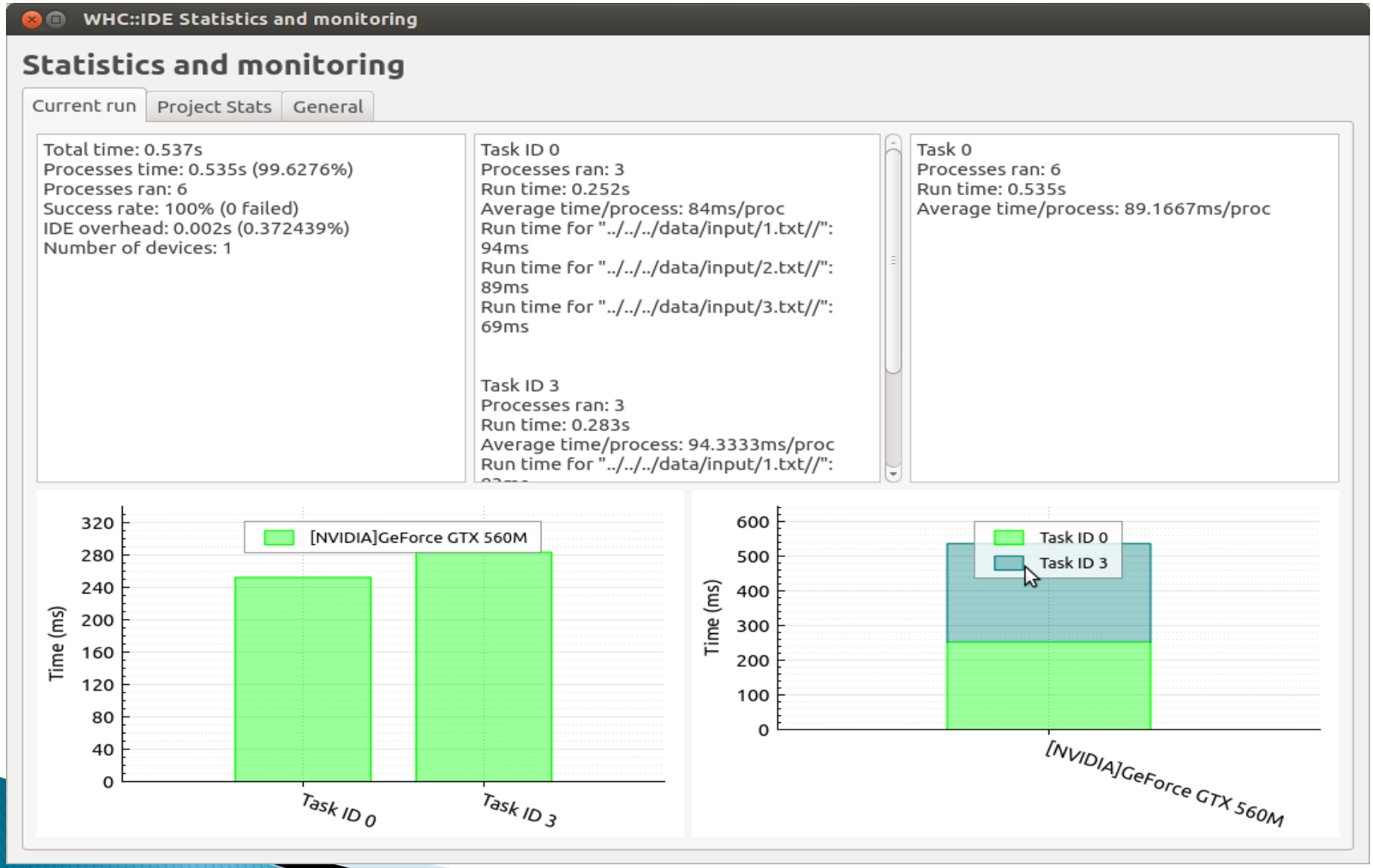
- Save file buffer flushed after every write
- Restoring consists of reading the file and continuing where the execution crashed

Logging and Stats

- The monitoring system connects to the execution signals
- Starts a timer for every process
- Starts a timer for the project

- Saves execution times
- Saves exit statuses and other info provided by Qprocess
- Graphical visualization of statistics

Logging and Stats



Execution Workflow

- Added cyclic graph detection
- Made possible to run different task nodes in parallel
- Replaced old topological sort based on DFS with new method of extracting nodes with 0 dependencies on each step

Editor

- Added autoindent
- Fixed many options not working
- Made it more stable (found the cause of some crashes)
- Brace matching

- Fixed highlighting/ auto-completion
- Made auto-completion aware of context