Contribution to ATLAS Computing in Algeria

Ecole Nationale Supérieure d'Informatique (ESI)

ECOLE NATIONALE
SUPÉRIEURE
D'INFORMATIQUE

Motivation of the collaboration

- First collaboration
  - a master student from ESI visiting CERN
  - worked on the ATLAS Distributed Computing systems

- Help ATLAS to face future computing challenges during the HL-LHC upgrade.

- Technical Associate Institute (ESI) is the framework in which this collaboration takes place.

Two main projects

- Porting of ATLAS software to parallel architectures
- Monitoring of conditions database access

Porting of ATLAS software to parallel architectures

- ATLAS expects to make substantial use of the next-generation HPCs; mixture of CPUs and accelerators, particularly GPUs.
- How to identify pieces of code that should be accelerated on GPUs ?
- **Objective:** leverage machine learning techniques to help in profiling code and determine whether it should be run on GPU
- *Students working on this topic:*
  - *One PhD student (Yacine Hakimi)*
  - *One research engineer*
  - *Six new master students*

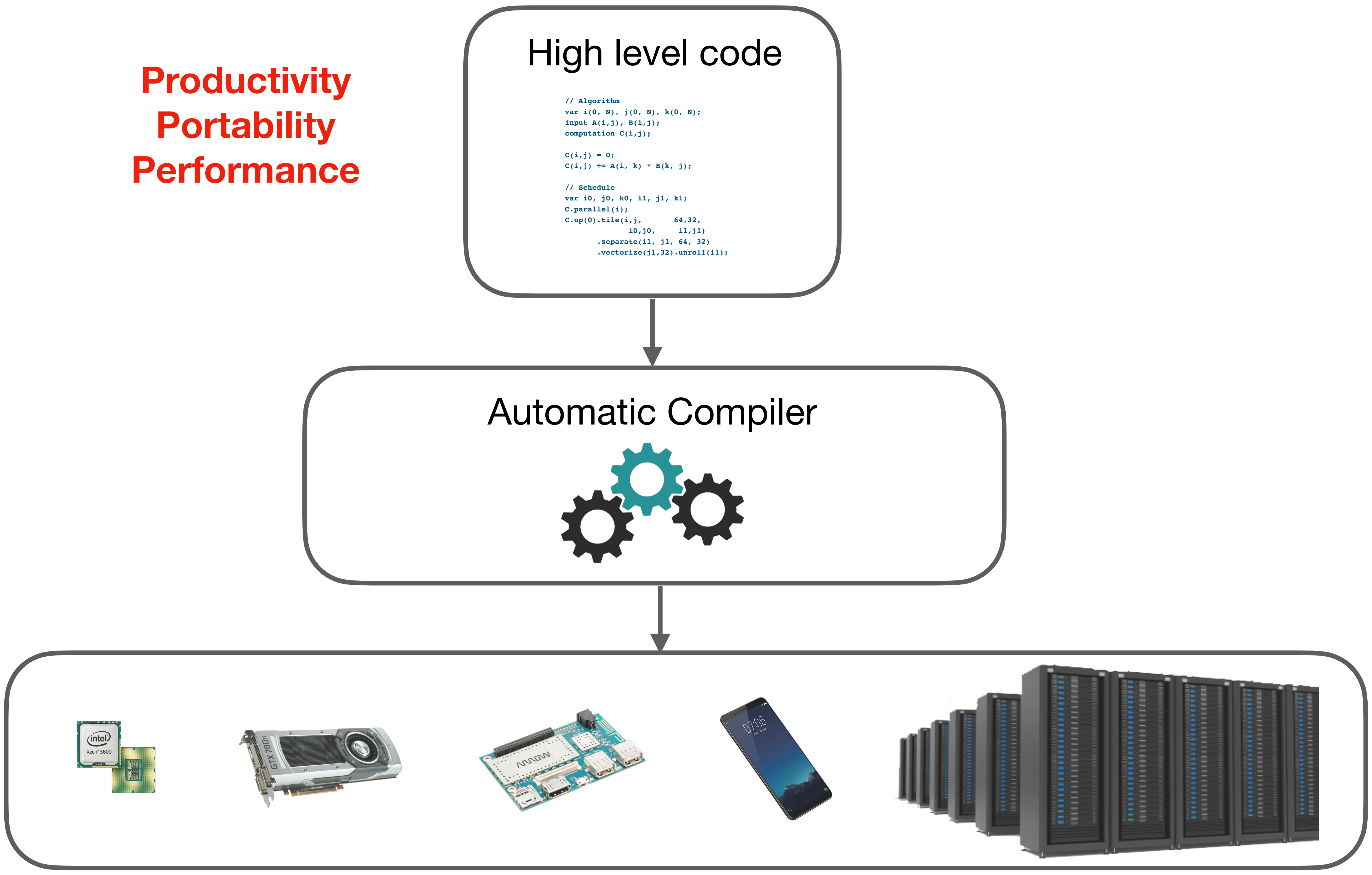
Monitoring of conditions database access

- Researchers are accessing the ATLAS experiment databases via ATLAS Distributed Computing systems (ADC systems)
- Many distributed agents have been set up to produce valuable information and metadata on the health and operation of ADC systems
- **Objective:** produce a general service capable of gathering difference pieces of information including logging, monitoring and performance information in order to analyse it, improve the operation of systems and detect any anomaly in these systems
- *Si Amer Mellissa Master Intern worked on this topic. She developed a platform based on elastic search to detect anomalies in database access*

Porting of ATLAS software to parallel architectures

Goal: Enable Compilers to Automatically Optimize Code

Productivity
Portability
Performance



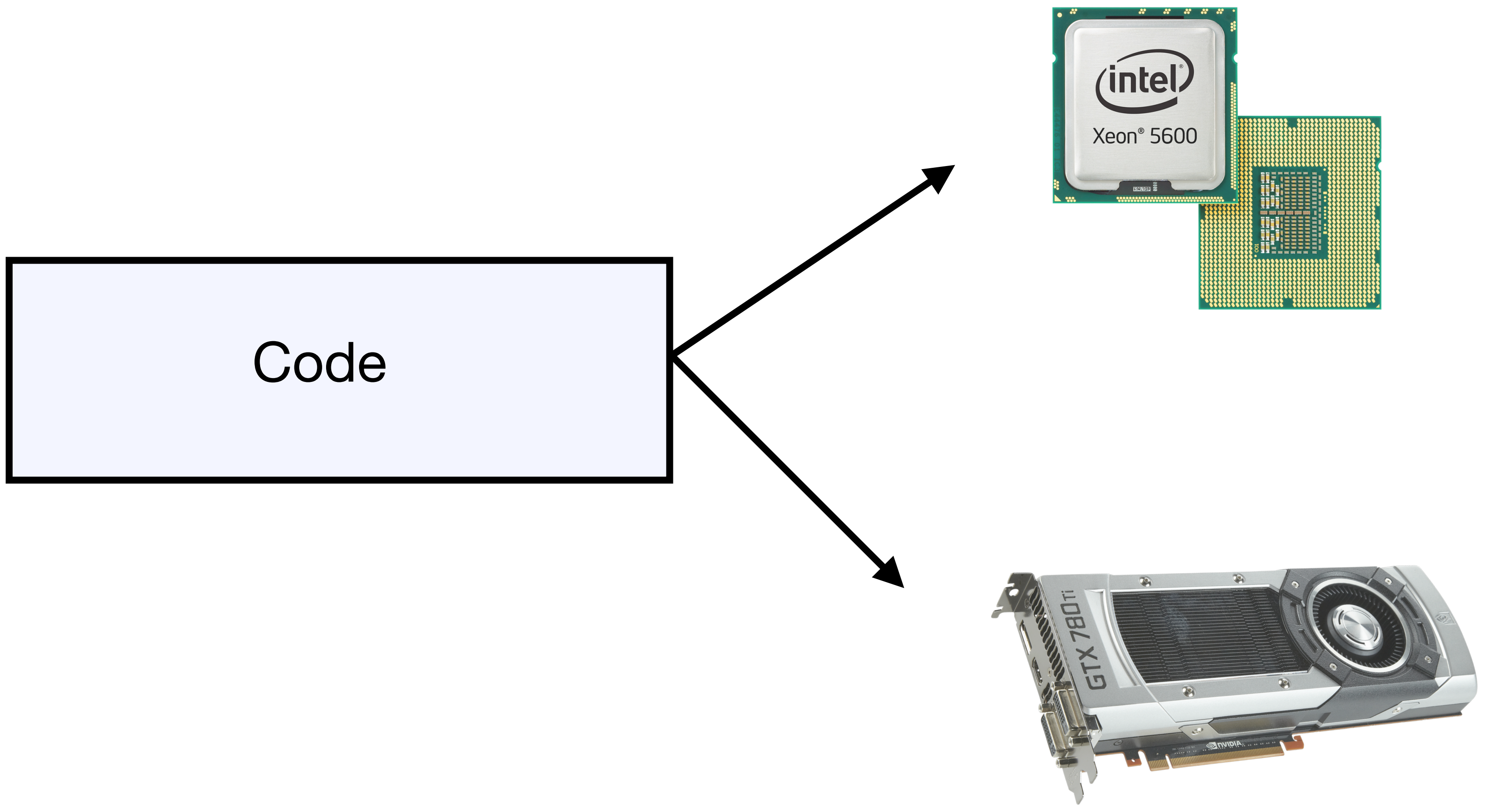
```
// Algorithm
var i(0, N), j(0, N), k(0, N);
input A(i,j), B(i,j);
computation C(i,j);

C(i,j) = 0;
C(i,j) += A(i, k) * B(k, j);

// Schedule
var i0, j0, k0, i1, j1, k1;
C.parallel(i);
C.up(0).tile(i,j,       64,32,
              i0,j0,     i1,j1)
       .separate(i1, j1, 64, 32)
       .vectorize(j1,32).unroll(i1);
```

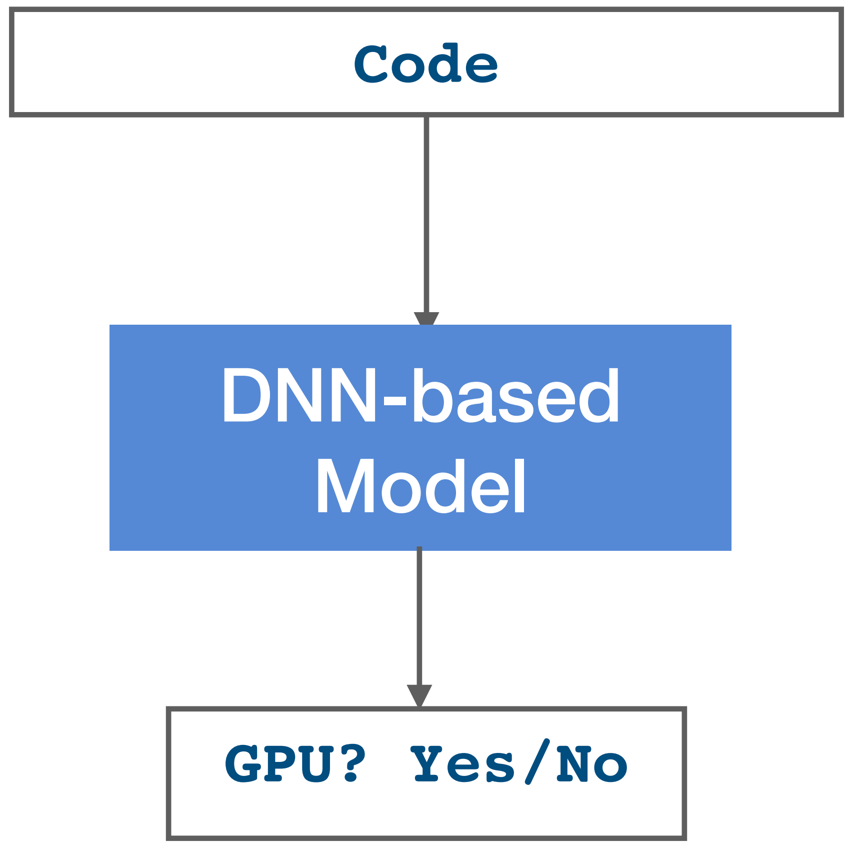
Example: Run a Code on CPU or GPU?

Decide whether to run an OpenCL code on **CPU** or on **GPU**



Our Solution

Build a deep learning model to help a compiler decide whether to map a code to GPU



Results

- We built a model that achieves state-of-the-art results
- Accuracy: 92%
- Paper
  - “Deep Learning and Classical Machine Learning for code mapping in Heterogeneous Platforms", 5th International Conference on Networking and Advanced Systems (ICNAS 2021), Annaba, Algeria, Nov. 2021