Towards as Strategy for Performance Prediction on Heterogeneous Architectures

Silvio Stanzani, Raphael Cóbe, Jefferson Fialho, Rogério Iope, Marco Gomes, Artur Baruchi and Júlio Amaral

Núcleo de Computação Científica – UNESP

VECPAR 2018

Agenda

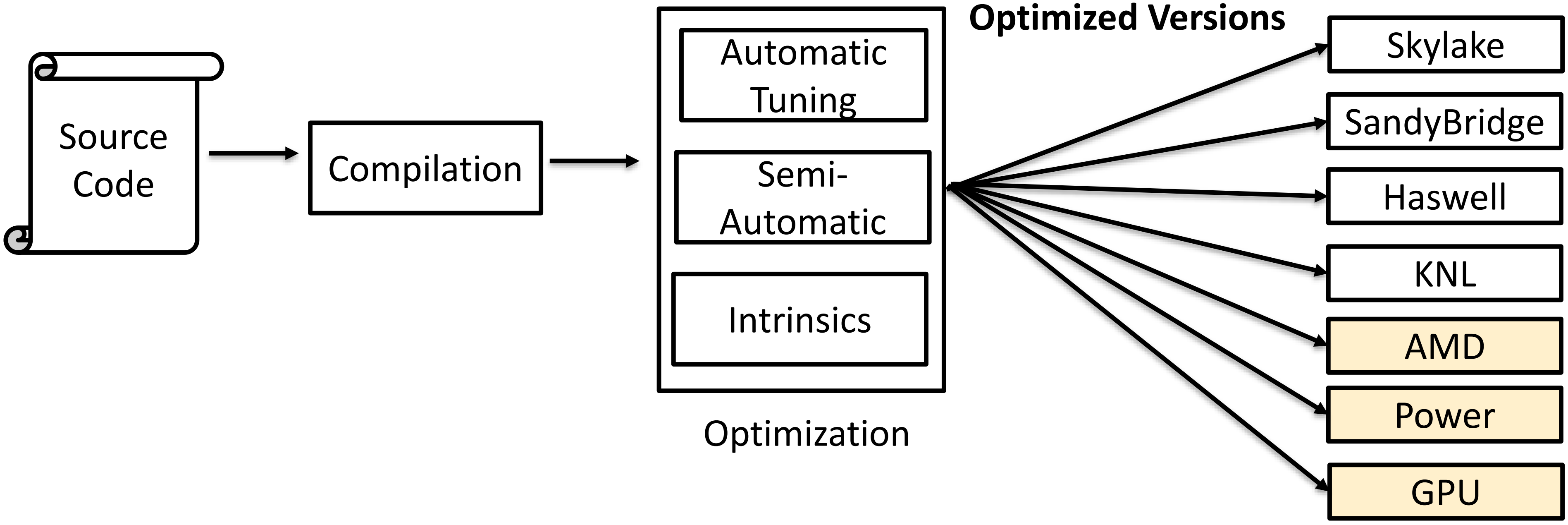
- NCC
- Heterogeneous Architectures
- Performance Prediction
- Strategy
- Evaluation
- Results
- Conclusion

UNESP Center for Scientific Computing

- Consolidates scientific computing resources for São Paulo State University (UNESP) researchers
  - It mainly uses Grid computing paradigm
    - ❑ Main users
      - ○ UNESP researchers, students, and software developers
      - ○ SPRACE (São Paulo Research and Analysis Center)
      - ○ Physicists and students
- Scientific Collaboration with
  - Caltech, Fermilab, CERN
  - São Paulo CMS Tier-2 Facility
- Partnership with Industry
  - Intel and Huawei

Heterogeneous Architectures

- Heterogeneous architectures presents several opportunities to improve performance;



How to choose the architecture that presents best performance for each application?

Outside of the scope of this work 🟨

Performance Prediction

- **Code Profiling**
  - Identify all the aspects of an application that can improve the performance targeting a specific architecture
- **Performance prediction**
  - Performance prediction based on small kernels
  - Performance characterization based on simulations
    - ❑ Compass Framework
  - Performance prediction based on regression model
- **Our approach**
  - Performance prediction to Support Runtime decisions:
    - ❑ Source code is not available;
    - ❑ Time limit constraint;
    - ❑ More than one optimized version for the same source code

Strategy

- The strategy inputs:
  - Application:
    - ❑ One binary code or more than one optimized version;
  - Set of architectures (Different Intel Generations):
    - ❑ Intel Xeon: Sandy Bridge, Haswell, Skylake
    - ❑ Intel Xeon Phi: Knights Landing
  - Measurements:
    - ❑ Ratio of giga floating-point operations per second (**GFLOPS**);
    - ❑ Ratio of giga floating-point operations by data transfer; Arithmetic Intensity (**AI**)
    - ❑ Clockticks per Instructions Retired (**CPI**).

Strategy

- Data Analysis:
  - Apply Estimated Processing Capacity (EPC) for each architecture

$$EPC(Application, Architecture) = \frac{GFLOPS * AI}{CPI}$$

  - Returns the rank position of each architecture according to absolute value of EPC

Evaluation

- Workload
  - A matrix multiplication (Intel Intrinsics);
  - A numeric model in finance (AVX-512 Exponentials and Reciprocals);
  - A N-Body simulation (Vectorized);
  - A Diffusion simulation (Scalar).
- Hardware

Architecture	Processor	Cores	Threads	Dram
SandyBridge	2x 2.6GHz	8	32	64GB
Haswell	2x 2.3GHz	36	72	128 GB
Skylake	2x 2.1GHz	48	96	192 GB
Knights Landing (cache mode)	1x 1.4GHz	68	272	192 GB
Knights Landing (flat mode)	1x 1.4GHz	68	272	192 GB

Results

Numeric Model in Finance

Architecture	Skylake	Haswell	SandyBridge	KNL (FlatMode)	KNL (Cache Mode)
Execution Time (Seconds)	458	1036	3443	235	224
Rank	3	2	1	4	5

Diffusion

Architecture	Skylake	Haswell	SandyBridge	KNL (Flat Mode)	KNL (Cache Mode)
Execution Time (Seconds)	511	309	220	425	1557
Rank	3	4	5	2	1

N-Body

Architecture	Skylake	Haswell	SandyBridge	KNL (Flat Mode)	KNL (Cache Mode)
Execution Time (Seconds)	306	467	1253	343	347
Rank	5	4	3	1	2

Matrix Multiplication (Intrisincs)

Architecture	Skylake	Haswell	SandyBridge	KNL (Flat Mode)	KNL (Cache Mode)
Execution Time (Seconds)	172	159	344	132	227
Rank	4	5	1	3	2

Wrong Prediction !

Conclusions

- This work presented a strategy to rank architectures according to performance gains for a given application.
  - It can be helpful for scheduling and run time decisions
  - low overhead (1 minute)

- Future work
  - Extend the metrics used to increase accuracy.
  - Evaluate with other architectures

Thanks!

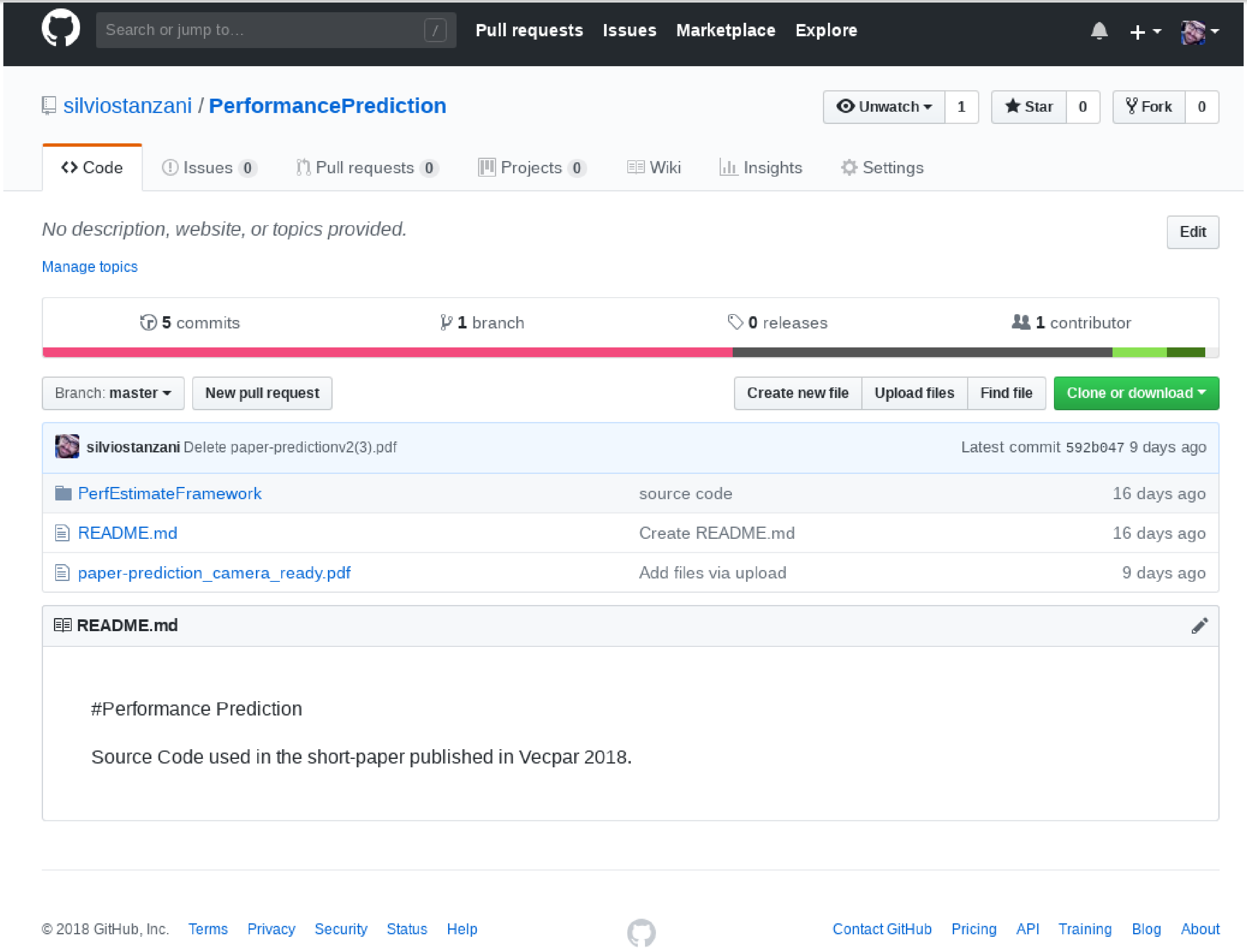
We would like to acknowledge Intel for the support !

Paper, Slides and source code:

https://github.com/silviostanzani/PerformancePrediction

Questions?

Performance Prediction - Github



silviostanzani / **PerformancePrediction**

No description, website, or topics provided.

silviostanzani Delete paper-predictionv2(3).pdf — Latest commit `592b047` 9 days ago

PerfEstimateFramework	source code	16 days ago
README.md	Create README.md	16 days ago
paper-prediction_camera_ready.pdf	Add files via upload	9 days ago

README.md

#Performance Prediction

Source Code used in the short-paper published in Vecpar 2018.

Performance Prediction

- Heterogeneous architecture has become more complex in scale, heterogeneous cores and memory system
- Profiling is essential to guide optimization and to support performance prediction.
  - Tipically focused on one architecture.
- How to compare the performance across different architectures?

Related Work

- Analysis based model based on simulation:
  - Performance characterization of OpenACC applications using simulations

- Prediction based on regression model of executing small part of the program
  - Accurate characterization

- COMPASS framework

- The difference of our approach
  - Simple strategy focusing runtime decisions
  - Not so accurate
  - Capable of be executed with time constraints