2nd International Workshop on Sustainable Software Engineering (SUSTAINSE)

GPPT: A Power Prediction Tool for CUDA Applications

GARGI ALAVANI, JINEET DESSAI & SANTONU SARKAR

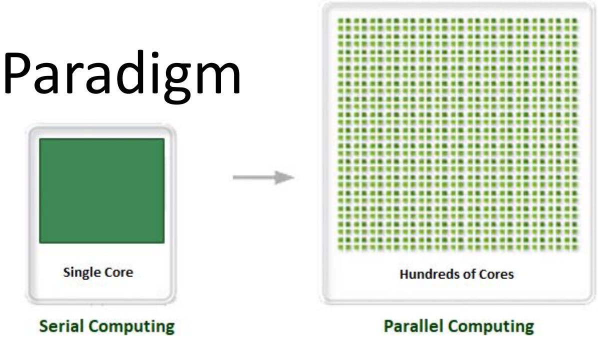
BITS PILANI K. K. BIRLA GOA CAMPUS, GOA INDIA

Outline

- Problem statement
- Our Contribution
- Tool architecture & implementation
- Results
- Analysis
- Tool Demo
- Conclusion

Introduction

- Shift in computing Paradigm



GPUs are part of commodity-level machines as well as supercomputers

- Performance gain is at the cost of power consumption

Predicting the power consumption of a CUDA kernel can help developers

- Understand power consumption of different program elements
- Refactor code to make it power efficient

Our Contribution

Presented GPPT, a power prediction tool built as an Eclipse plugin

GPPT predicts power consumption of an application without the need of running it

GPPT works on Windows OS and is tested on three GPU architectures: Tesla, Maxwell, Volta

Tool generates a report on features contribution to the power consumption of an application using LIME

Existing Approaches

Counter Based Models

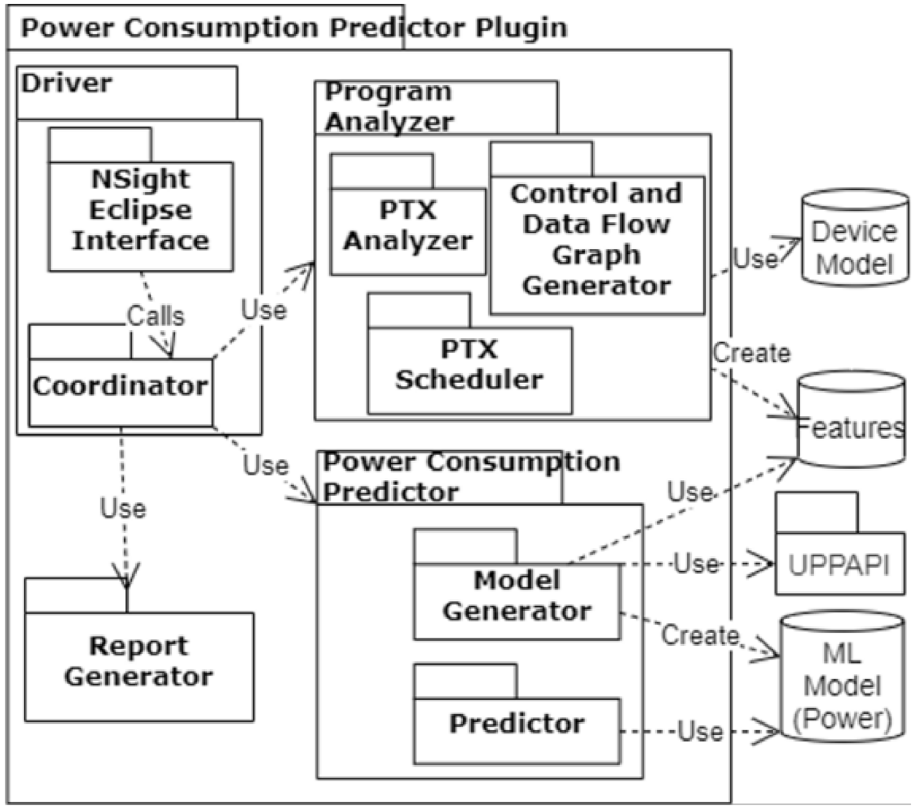
- Uses Hardware counters
- Multiple runs of an application are sometimes required to collect this data.
- Some hardware counters may not be available which affects the prediction model [Nagasaka et al.]
- Some architectures allow counters to a whole SM

Static Input Based Models

- Not as explored as counter based models
- Work by Zhao et al.
  - Gives power prediction for the GPU by counting each PTX instruction as input.
- Work by Hong et al.
  - Analytical integrated power and performance prediction model, fails to predict asymmetric and control-flow intensive applications.

- Mittal and Vetter et al. survey on GPU Energy Efficiency concluded that there is a need for using multiple approaches at the chip design level, architectural level, programming level, etc. , to get the maximum increase in GPU energy efficiency.
- A survey by Bridges et al. suggest that a PTX-based power model can produce valuable and informative predictions when one wants to optimize an application.

Tool Architecture



Program Analyzer

Consists of three submodules

- PTX Analysis
- Control and Data Flow Generator
- PTX Scheduler

Power Consumption Predictor

- ML Model: Random Forest Regressor and XGBoost Regressor
- Power is collected using UPPAPI

Report Generator

- LIME generates detailed report on features contributing to prediction

Features utilized in ML model

Feature	Source
Compute Capability	User
Number of threads per Block	User
Number of computing instructions	PTX Analyzer
Number of global memory instructions	PTX Analyzer
Number of Simulated global memory instructions	PTX Scheduler
Number of shared memory instructions	PTX Analyzer
Number of shared memory instructions	PTX Scheduler
Number of Miscelleneous instruction	PTX Analyzer
Number of Simulated Miscelleneous instruction	PTX Scheduler
Number of instruction issue cycles per SM	PTX Analyzer
Occupancy	User

Power Consumption Predictor Results

Model Hyperparameters

Model	Hyperparameters
Random Foresr	random_state=7, max_features='auto', n_estimators= 400, max_depth=14, min_samples_split=2, criterion='mse'
XGBoost	colsample_bytree=0.7, learning_rate=0.05, n_estimators=450, max_depth=9, min_child_weight=3, silent=1, subsample=0.7

Model validation Metrics Score

Regression Technique	R^2 score	RMSE	MAE
Random Forest Regressor	0.9128	8.2629	4.03048
XGBoost Regressor	0.9309	7.3272	3.2978

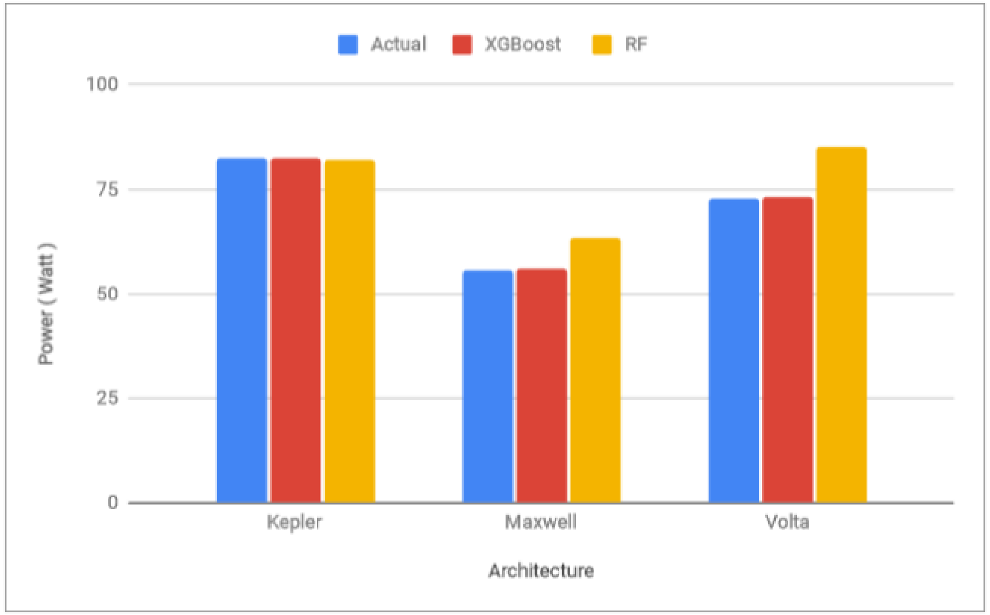
Feature Importance

Feature	RF Regressor	XGBoost Regressor
occupancy	0.01013	0.0237
shar_inst_kernel	0.01031	0.1376
block_size	0.01352	0.0122
shar_inst_sim	0.01357	0.04653
glob_inst_sim	0.04191	0.05
misc_inst_sim	0.06951	0.08019
glob_inst_kernel	0.06994	0.07293
comp_inst_kernel	0.11184	0.09712
compute_capability	0.1288	0.2288
misc_inst_kernel	0.1324	0.12608
inst_issue_cycles	0.3978	0.12463

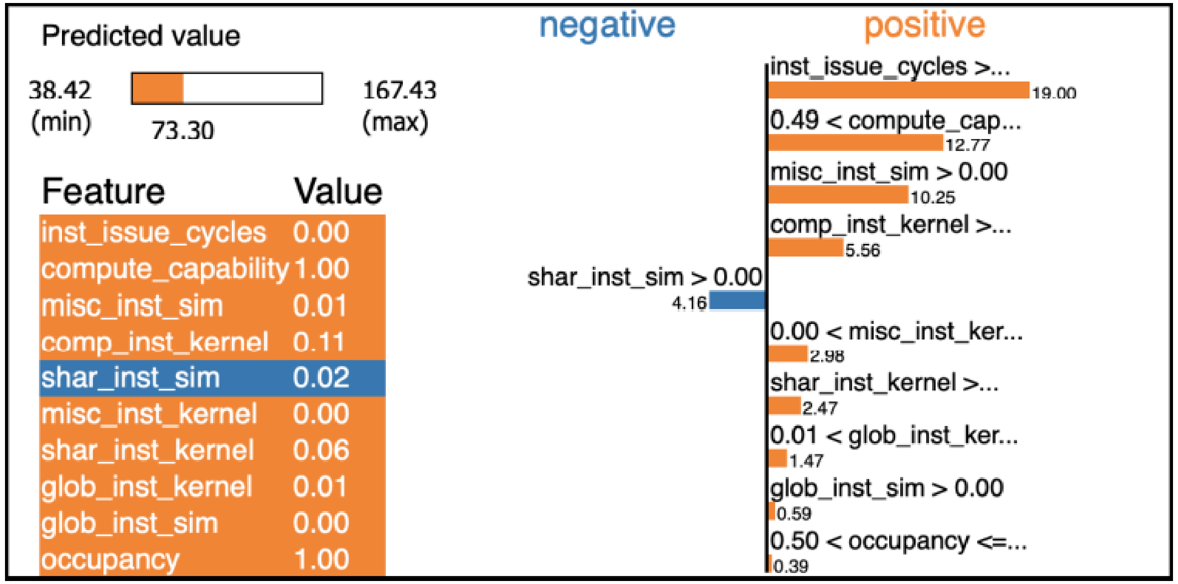
Case Study: Matrix Multiplication

Result across architecture

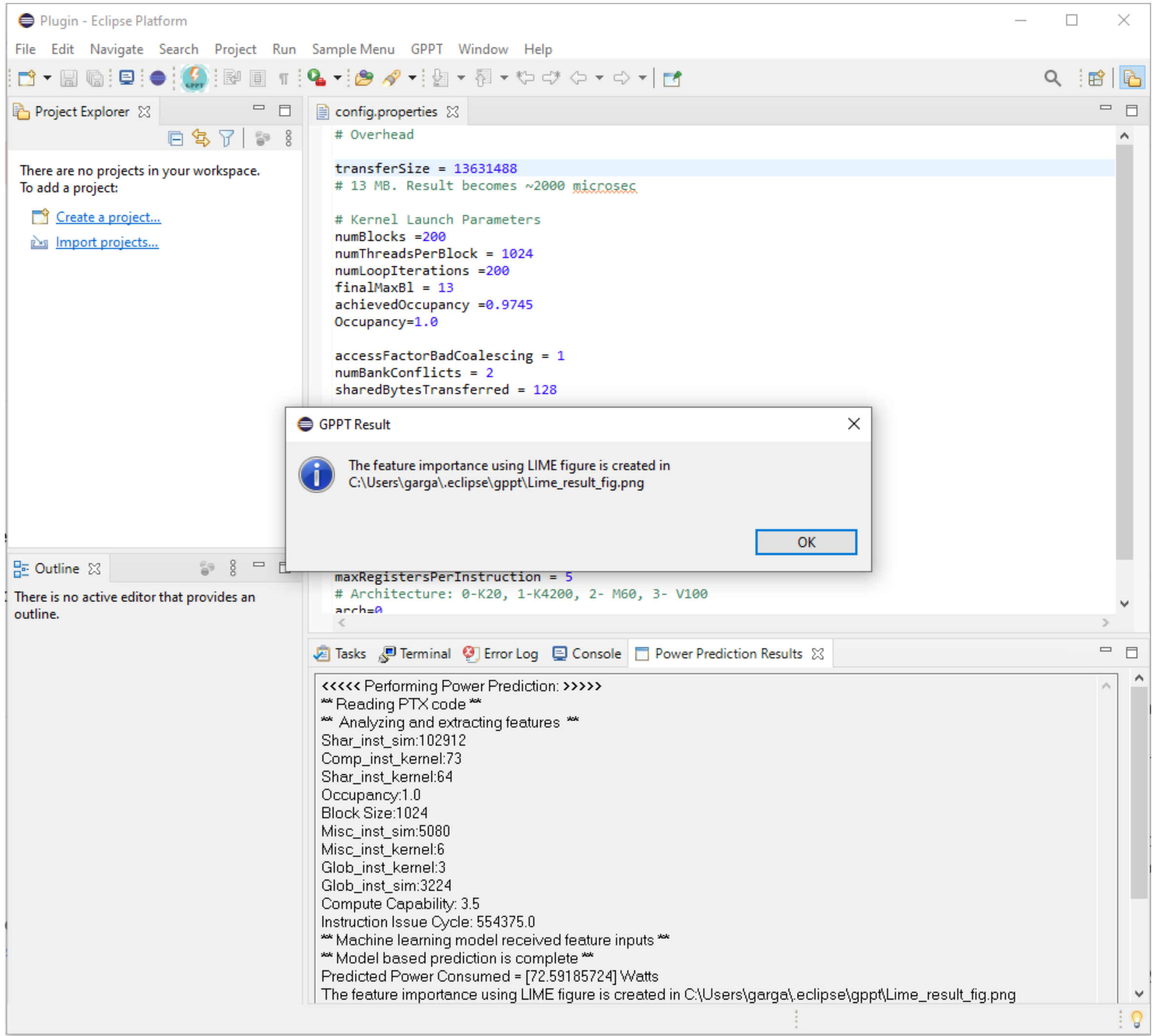
GPU Architecture	Actual	XGBoost	RF
Kepler	82.2078 W	82.40907 W	81.911 W
Maxwell	55.85302 W	56.19047 W	63.74 W
Volta	73.13773 W	73.29901 W	84.92 W



LIME Report



Tool Demo : Plugin in Eclipse IDE

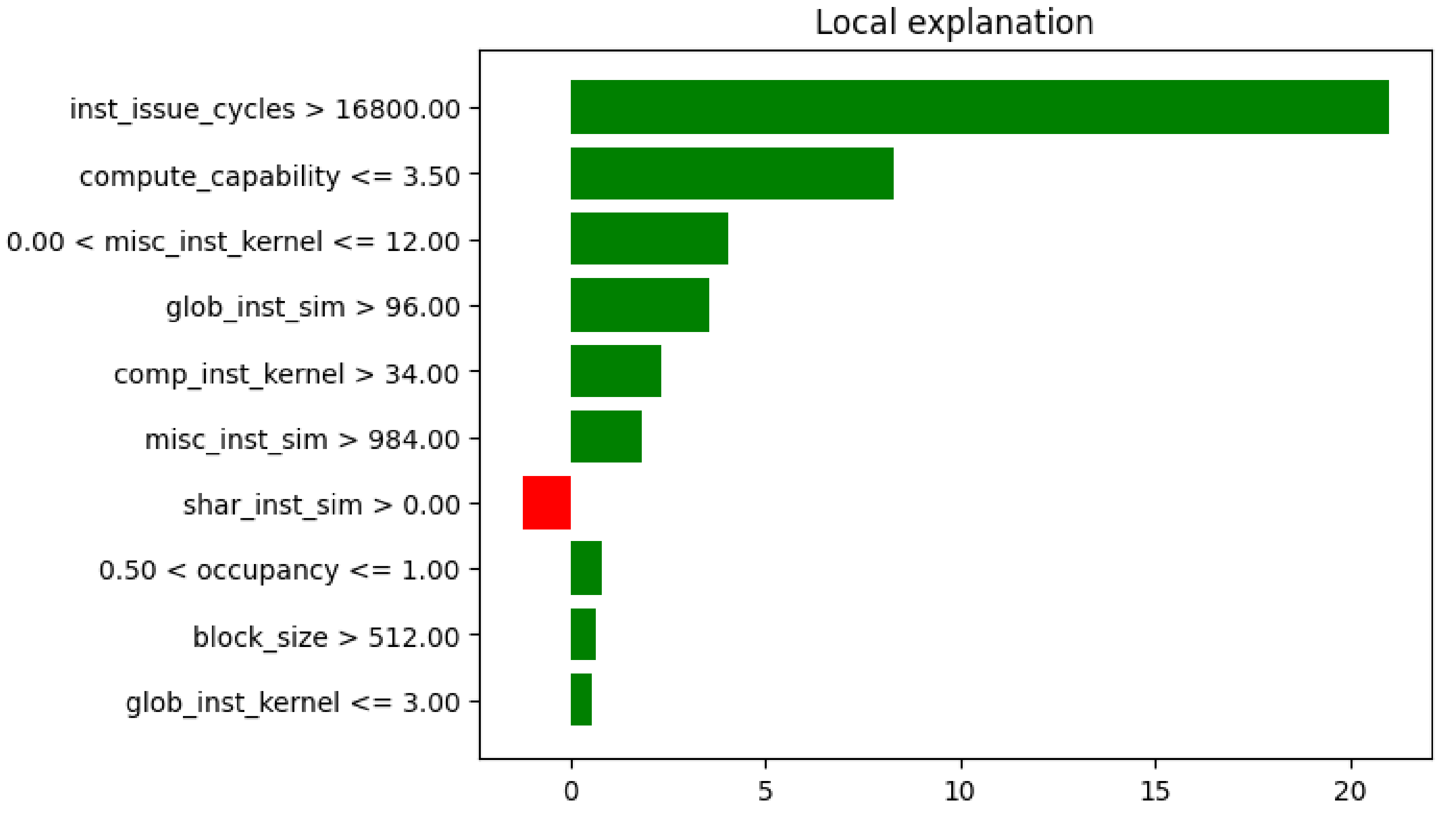


Tool Demo : Result Generated

Tasks | Terminal | Error Log | Console | Power Prediction Results

```
<<<<< Performing Power Prediction: >>>>>
** Reading PTX code **
**  Analyzing and extracting features  **
Shar_inst_sim:102912
Comp_inst_kernel:73
Shar_inst_kernel:64
Occupancy:1.0
Block Size:1024
Misc_inst_sim:5080
Misc_inst_kernel:6
Glob_inst_kernel:3
Glob_inst_sim:3224
Compute Capability: 3.5
Instruction Issue Cycle: 554375.0
** Machine learning model received feature inputs **
** Model based prediction is complete **
Predicted Power Consumed = [72.59185724] Watts
The feature importance using LIME figure is created in C:\Users\garga\.eclipse\gppt\Lime_result_fig.png
```

Tool Demo : LIME report



Conclusion

We presented GPPT, an eclipse plugin tool that analyses a GPU application and predicts its power consumption, along with a detailed report on its prediction.

GPPT employs program analysis and simulation of PTX code using java based applications.

Generated features are supplied to machine learning model. Random Forest and XGBoost Regressor demonstrate high accuracy of 0.91 and 0.93 R^2 score.

Future work can involve including some hardware-specific features which can be computed using vendor-supplied information

Download GPPT

https://gargialavani.github.io/gppt/index.html

THANK YOU