LEAPER: Fast and Accurate FPGA-based System Performance Prediction via Transfer Learning

Gagandeep Singh, Dionysios Diamantopoulos, Juan Gómez-Luna, Sander Stuijk, Henk Corporaal, and Onur Mutlu
Executive Summary

**Background:** Machine learning (ML)-based performance modeling has gained traction as a way to overcome the slow accelerator generation and implementation process on an FPGA.

**Problem:** Three key shortcomings of prior ML-based techniques:
- Models are trained for a specific environment
- Training requires large amounts of data
- Models trained using a limited number of samples are prone to overfitting

**Goal:** Overcome limitations of traditional ML-based techniques to provide accurate and fast prediction of performance and resource usage of accelerator implementation on an FPGA.

**Our contribution:** LEAPER, a transfer learning-based approach for prediction of performance and resource usage for accelerator implementation on an FPGA:
- Transfer ML-based model from edge to cloud platforms
- Transfer ML-based model across applications
- Provide fast and accurate predictions of previously unseen accelerator optimization options

**Key Results:** Evaluate LEAPER across 5 state-of-the-art cloud FPGA-based platforms with 2 different interconnect technologies on 6 real-world applications:
- Provides, on average, **85% accuracy** when we use our transferred model for prediction in a cloud environment
- Reduces design-space exploration time for accelerator implementation on an FPGA by **10x**, from days to only a few hours.
- Unlike state-of-the-art techniques, we show that classic non-neural network-based models are enough to build an accurate predictor to evaluate accelerator implementation on an FPGA.
Talk Outline

Motivation

LEAPER: Implementation

Evaluation of LEAPER and Key Results

Summary
Wide Adoption of FPGAs

• FPGAs provide a tradeoff between **programmability** and **efficiency**

• FPGAs being **deployed from edge to cloud** for many applications

Particle Physics

Atmospheric Modeling

Genome Sequencing

Image sources: http://www.flometrics.com/fluid-dynamics/computational-fluid-dynamics
Naoe, Kensuke et al. "Secure Key Generation for Static Visual Watermarking by Machine Learning in Intelligent Systems and Services" IJSSOE, 2010
The Key Problem

Accelerator Implementation Process

Application: C/C++/System C

- Large Optimization Space
- Time Consuming Process (days or months)

- High-level Synthesis (HLS)
- Logic Synthesis
- Technology Mapping
- Place and Route
- Bitstream

HLS Estimates

Machine Learning

Implementation Report: Performance/Timing, Resource Usage

Low Productivity
Traditional ML-Based Approach

Low-end FPGA
- Fast bitstream generation
- Cheap
- Easily accessible

High-end FPGA
- Slow bitstream generation
- Expensive
- Not easily accessible

Trained for Specific Environment

SAFARI
Overcome limitations of traditional ML-based techniques to provide accurate and fast prediction of performance and resource usage of accelerator implementation on an FPGA.
Our Proposal

LEAPER

Transfer learning-based approach for prediction of performance and resource usage in an FPGA-based system
Transfer Learning

- Transfer knowledge **from previous experiences to solve new tasks**

- Similar to humans, algorithms can learn from experiences

- **Rather than learning from scratch**
LEAPER

Low-end FPGA
- Fast bitstream generation
- Cheap
- Easily accessible

High-end FPGA
- Slow bitstream generation
- Expensive
- Not easily accessible

Data collection:
- Design of experiments
- Brute force

Fast design-space exploration
LEAPER: Implementation

**Base Model Building**

- Application
- LLVM Kernel Analyzer
  - Instruction Mix
  - Memory Behavior
  - ILP
- Accelerator Generation
  - Accelerator Implementation
  - FPGA Deployment
  - Accelerator Features
- Hyperparameter Tuning
- Training Dataset
- Base Learner
- Model Generation
- Low-end Edge Model
  - Performance/Resource Utilization

**Target Model Building**

- Application
- LLVM Kernel Analyzer
  - Instruction Mix
  - Memory Behavior
  - ILP
- Accelerator Generation
  - Accelerator Implementation
  - FPGA Deployment
  - Accelerator Features
- Few-shot Learning Dataset
- High-end Cloud Model
- Performance/Resource Utilization
- Ensemble Transfer Learner
- Model Generation
Base Model Building
Phase 1: LLVM Analyzer

**Base Model Building**

- **Application Features**
  - Instruction mix
  - ILP
  - Reuse distance
  - Memory traffic
  - Register traffic
  - Memory footprint

**LLVM Kernel Analyzer**
- Instruction Mix
- Memory Behavior
- ILP

**Accelerator Generation**
- Accelerator Implementation
- FPGA Deployment

**Training Dataset**
- Hyperparameter Tuning
- Base Learner
- Model Generation
- Low-end Edge Model
  - Performance/Resource Utilization

**Evaluation Dataset**
- Few-shot Learning Dataset
- High-end Cloud Model
- Model Generation
  - Ensemble Transfer Learner
  - Performance/Resource Utilization
Phase 2: Accelerator Generation

Design of experiments technique to **minimize the number of experiments** while data collection.
Phase 3: Base Model Training

**Application Features**
- Instruction mix
- ILP
- Reuse distance
- Memory traffic
- Register traffic
- Memory footprint

**Accelerator Optimization Options**
- Loop pipelining
- Loop unrolling
- Array partitioning
- Inlining
- Dataflow
- Burst read/write
- FPGA frequency

**Diagram:**
- Application
- Instruction Mix
- Accelerator Generation
- Accelerator Implementation
- FPGA Deployment
Target Model Building via Transfer Learning
Phase 1: LLVM Analyzer

Target Model Building
Phase 2: Accelerator Generation

Design of experiments technique to minimize the number of experiments while data collection
Phase 3: Target Model Training

Create a **few-shot learning dataset** to learn the change in distribution for the new environment (application/hardware platform)
Phase 3: Target Model Training

To transfer a model, LEAPER uses:

- **Few-shot learning dataset** to train an ensemble of transfer learners
- **Transfer learner** to perform a non-linear transformation of predictions from the base model to the target model
Talk Outline

Motivation

LEAPER: Implementation

Evaluation of LEAPER and Key Results

Summary
Evaluation Methodology (1/2)

• **Goal:**
  1. Transfer ML-based model **from edge to cloud platforms**
  2. Transfer ML-based model **across applications**
  3. Predictions of previously **unseen accelerator optimization options**

• **Nimbix cloud** as the **target high-end platform** with:
  - 5 FPGA configurations
  - 2 CAPI-based interconnects (CAPI1/CAP2)

• **PYNQ-Z1 ZYNQ** as the base **low-end platform SAFARI**
Evaluation Methodology (2/2)

• 6 real-world workloads:
  - Image processing
    • Histogram (HIST)
    • Canny edge detection (CEDD)
  - Machine learning
    • Binary long short-term memory (BLSTM)
    • Digit recognition (DIGIT)
  - Databases:
    • Relational operation (SELECT)
    • Stream compaction (SC)

• Programming tools:
  - Xilinx design tools (Vivado and HLS)
  - IBM CAPI-SNAP Framework
Performance Prediction: Transfer From Edge to Cloud

- Transfer from low-end edge PYNQ-Z1 board to high-end cloud FPGA-based systems
• Transfer from low-end edge PYNQ-Z1 board to high-end cloud FPGA-based systems

LEAPER can effectively transfer model from edge to cloud platform using only 5-10 samples

Reduces design-space exploration time by 10x than training from scratch (from days to only a few hours)
Performance Prediction: Transfer Across Applications

- Transfer **across applications on low-end edge PYNQ-Z1 board**
Performance Prediction: Transfer Across Applications

- Transfer **across applications on low-end edge PYNQ-Z1 board**

**LEAPER** can effectively transfer models across applications with on average **85% accuracy**

---

**Base Models: BLSTM**

<table>
<thead>
<tr>
<th>Target Model</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEDD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIGIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gmean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Prediction Comparison: Unseen Accelerator Optimizations

- Prediction of **previously unseen accelerator optimization options** on the base platform
- **Comparison with three popular ML-based techniques:** XGBoost (XGB), artificial neural network (ANN), and decision tree (DT)
Prediction Comparison: Unseen Accelerator Optimizations

• Prediction of **previously unseen accelerator optimization options** on the base platform

• **Comparison with three popular ML-based techniques:** XGBoost (XGB), artificial neural network (ANN), and decision tree (DT)

**LEAPER** provides both **high accuracy and sample-efficiency** compared to other ML-based techniques

![Graph showing performance metrics for LEAPER, XGB, ANN, and DT](image)
More in the Paper

• Accuracy analysis for **transferring resource usage models**

• **Time and cost analysis** to build ML models using LEAPER and traditional approach

• Transfer to a **wide range of cloud FPGA configurations and applications**

• Comparison to **different transfer learning algorithms**

• **Explainability analysis** of LEAPER

• **Discussion on limitations**
More in the Paper

- Accuracy analysis for transferring resource usage models
- Time and cost analysis to build ML models using LEAPER and traditional approach
- Transfer to a wide range of cloud FPGA configurations and applications
- Comparison to different transfer learning algorithms
- Explainability analysis of LEAPER
- Discussion on limitations

LEAPER: Fast and Accurate FPGA-based System Performance Prediction via Transfer Learning

Gagandeep Singh\textsuperscript{a}  Dionysios Diamantopoulos\textsuperscript{b}  Juan Gómez-Luna\textsuperscript{a}
Sander Stuijk\textsuperscript{c}  Henk Corporaal\textsuperscript{c}  Onur Mutlu\textsuperscript{a}
\textsuperscript{a}ETH Zürich  \textsuperscript{b}IBM Research Europe, Zurich  \textsuperscript{c}Eindhoven University of Technology

LEAPER transfers previously trained models to predict the performance and resource usage of accelerator implementation.

LEAPER is cheaper (with 5-shot), faster (up to 10x), highly accurate (85%) at predicting performance and resource usage in a new environment than building model from scratch.
LEAPER:
Fast and Accurate FPGA-based System Performance Prediction via Transfer Learning

Gagandeep Singh, Dionysios Diamantopoulos, Juan Gómez-Luna, Sander Stuijk, Henk Corporaal, and Onur Mutlu