**DDARING: Dynamic Data-Aware Reconfiguration, INtegration & Generation**

V. Sarkar¹, T. Conte¹, J. Shirako¹, R. Vuduc¹, D. Chen², W.-M. Hwu², S. Mahlke³, V. Prasanna⁴

¹Georgia Tech, ²University of Illinois, ³University of Michigan, ⁴University of Southern California

**Distribution Statement A – Approved for Public Release, Distribution Unlimited.**

**This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).**

**The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.**

---

**Specialized Functions: Software Defined Hardware (SDH)**

**Input Datasets**
- Ingest
- Map
- Analyze

**Output Datasets**
- Output

---

**Overview**

**DDARING Software Tool Chain (TA2)**

- Intrepydd Programming Model (IPM)
- Static Data-aware Optimizer (SDO)
- Dynamic Kernel Reoptimization (DKR)
- Auto-tuning & Reconfiguration (AR)

**Knowledge Base:** Kernels, Data Access Patterns, Configurations (KB)

---

**Approach**

**Programming Model (PM)**

- **Add high-level primitives to current Intrepydd language**
  - Primitives include map, partition, reduce, etc.
  - Simplify programming and high-quality parallel code generation

- **For example, the following map operation:**

  ```
  c * map(\(y, \{a, b, c, d, e, f, g, h, i\}\) => \((a+b+c+c)+d+e+f+g+h+i\));
  ```

  **Can be translated to:**

  ```
  c + map(y => x => c + x, [a, b, c, d, e, f, g, h, i]);
  ```

**Static Data-aware Optimizer (SDO)**

- **Generate all variants and store them in a library**
- **Use input data + knowledge base to select best variant at runtime**
- **Record runtimes in knowledge base for future use**

**Dynamic Kernel Reoptimization (DKR)**

- **Best programmer effort**
  - Use input data + knowledge base to select best variant at runtime

**Auto-tuning & Reconfiguration (AR)**

- **Lightweight run-time of DDARING that interfaces to the specifics of the TA1 architecture.**
  - Profiling is input into the trained LSTM to detect phase changes during dynamic execution.
  - When a new phase is detected, it is looked up in a code & configuration cache (CCC) where cache misses from this are handled by the knowledge base (which in turn was populated by the auto-tuner).

---

**Answers to Heilmeier Questions 1 to 4**

1. **Goals:**
   - New software tool chain to accelerate data-intensive workflows with near-ASIC performance, and programmability of Python
   - Dynamically reconfigure hardware to match data and algorithm requirements

2. **Current practice:**
   - H/w reconfiguration is orders of magnitude slower than 1μs
   - Tool chains do not dynamically reconfigure hardware nor exploit data characteristics
   - Programmability of current tool chains for accelerators is only accessible to “ninja” experts

3. **Novelty:**
   - Dynamic data-driven co-optimization of algorithmic variants, hardware configurations and code transformations
   - Mapping of high-level Python-based Intrepydd programming model to reconfigurable hardware

4. **Impact:**
   - Near-ASIC performance for data-intensive algorithms with high programmability
   - Transition to open source/vendor products
   - Adoption by transition partners in DoD community

---

**Run & check**

- **Detect precision violations**
- **Integer: Hardware provides low cost detection**
- **Floating-point: Use application-level error metrics for detection**
- **Trigger profiling and reconfiguration**

**Distribution Statement A – Approved for Public Release, Distribution Unlimited.**

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).

The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

**www.darpa.mil**

PI & POC: Vivek Sarkar
Email: vsarkar@gatech.edu

---

**V. Sarkar¹, T. Conte¹, J. Shirako¹, R. Vuduc¹, D. Chen², W.-M. Hwu², S. Mahlke³, V. Prasanna⁴**

¹Georgia Tech, ²University of Illinois, ³University of Michigan, ⁴University of Southern California

---

**Specialized Functions: Software Defined Hardware (SDH)**

---

**Knowledge Base (KB)**

- **Workflows**
- **Kernel/Task**
- **HW Configurations**

**Auto-Tuning & Reconfiguration (AR)**

- **Profile monitor**
- **Profile monitor**
- **Profile monitor**

**Code & Configuration Cache (CCC)**

- **Architecturespecific**
- **Code & Configuration Cache (CCC)**
- **Knowledge Base**
- **Auto-Tuner**

---

**Distribution Statement A – Approved for Public Release, Distribution Unlimited.**

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).

The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

**www.darpa.mil**

PI & POC: Vivek Sarkar
Email: vsarkar@gatech.edu