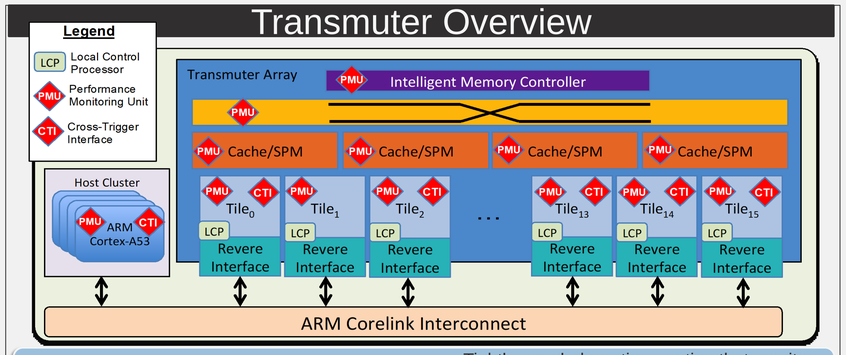
Transmuter – A Reconfigurable Computer

R. Dreslinski[1], T. Mudge[1], H.-S. Kim[1], D. Blaauw[1], C. Chakrabarti[2], M. O'Boyle[3], M. Cole[3]
[1]University of Michigan, [2]Arizona State University, [3]University of Edinburgh

Specialized Functions: Software Defined Hardware (SDH)

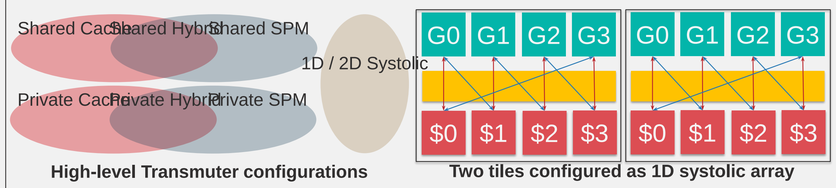
Transmuter Overview



- **Massively parallel** architecture employing a SIMD/MIMD paradigm and in-order Arm cores
- **Fast reconfigurable** (within 10s of cycles) supporting modes such as cache / scratchpad / queue, systolic array, etc.
- Tightly-coupled run-time routine that monitors hardware via i**ntrospection**
- **Intelligent memory controller** for better data marshaling to maximize off-chip bandwidth utilization

Hardware Design

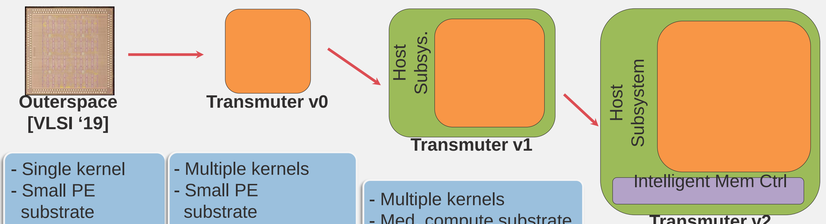
Architectural Configurations



High-level Transmuter configurations

Two tiles configured as 1D systolic array

Chip Development



- Single kernel
- Small PE substrate
- Cache-SPM reconfiguration

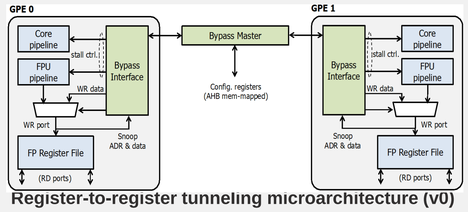
- Multiple kernels
- Small PE substrate
- R2R tunneling, systolic array, etc.

- Multiple kernels
- Med. compute substrate
- Host+software stack
- Revere interface

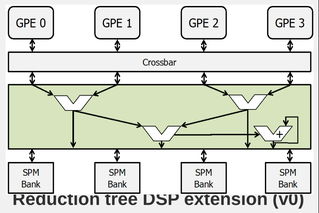
- Multiple applications
- Large compute substrate
- Malleable prefetcher
- Data packing and restructuring
- Off-chip DDR interface

	Bandwidth Eff. (M NNZ/GB)	Compute Density (M NNZ/s/mm^2)	Energy Eff. (M NNZ/J)
Min.	6.4 (8.4x) (58.3x)	0.3 (10.7x) (25.7x)	6.1 (8.4x) (5.4x)
Max.	15.5 (16.5x) (172.7x)	0.7 (22.1x) (78.3x)	8.4 (31.2x) (11.5x)
Mean	11.7 (11.7x) (77.6x)	0.5 (17.1x) (36.9x)	7.3 (12.6x) (8.4x)

Measured improvements over CPU/GPU for SpMM [VLSI 19]

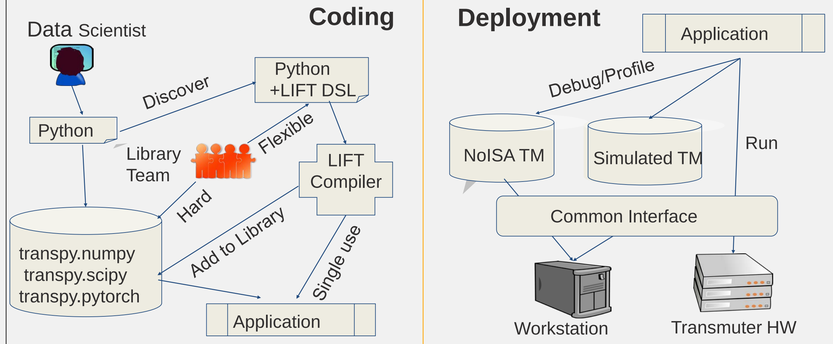


Register-to-register tunneling microarchitecture (v0)



Reduction tree DSP extension (v0)

Software Design



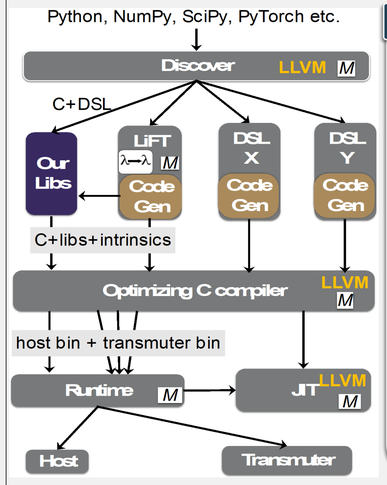
LIFT Compiler

```
s.indptr.slide(2, 1).map(lambda
s_indptrs:
    v.map(lambda row:
        zip(s.data[s_indptrs],
row[s.indices[s_indptrs]]).map(mult)
.reduce(add, 0)))
```

```
s.indptr.slide(2, 1).mapTile(lambda
s_indptrs:
    v.mapGPE(lambda row:
        zip(s.data[s_indptrs],
row[s.indices[s_indptrs]]).mapSeq
(mult).reduceSeq(add, 0)))
```

ML Driven Rewrite | Code Generator

```
int main(){
    float *v_initial_param_4_9 = reinterpret_cast<float*> GPEQ_POP();
    float *v_initial_param_5_10 = reinterpret_cast<float*> GPEQ_POP();
    float *v_user_func_23_12 = reinterpret_cast<float*> GPEQ_POP();
    int v_N_0 = GPEQ_POP();
    // For each tile
    for (int v_i_6 = 0; v_i_6 <= (-1+v_N_0/8); ++v_i_6) {
        int v_i_7 = GPEQ_POP();
        // For each element processed sequentially
        for (int v_i_8 = 0; v_i_8 < 2; ++v_i_8)
            v_user_func_23_12[v_i_8+2*v_i_7+8*v_i*6] =
                add(v_initial_param_4_9[v_i_8+2*v_i_7+8*v_i*6],
                v_initial_param_5_10[v_i_8+2*v_i_7+8*v_i*6]);
        LCPQ_PUSH(1); // Sync to LCP
    }
}
```

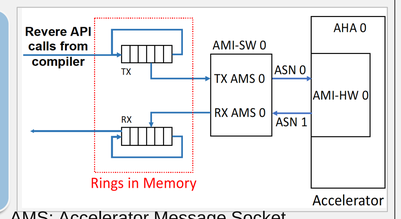


Multiple Paths to Success

- **Write in Python** with familiar library APIs
- **Discover idioms** implicitly representing promising applications of Transmuter algorithms
- **Exploit the best C** implementations from
  - Own ML-tuned libraries
  - Code generated by LIFT
  - Other imported DSLs
- **LIFT** DSL compiler provides optimization of search space traversal via safe rewrite rules
  - **Easy upgrade path** as the architecture develops – new feature -> add a new rule!
- Run-Time: **feedback loop controls reconfiguration** and JIT compilation, integrated with multi-versioning in our compiled code

Hardware/Software Interface

- Arm system architecture providing direct assignment of accelerators to software
- Uses industry-standard software stacks built on PCI and SR-IOV
- Revere interface is only used for **commands/control flow**
- Use DMA interface for **data transfers**



AMS: Accelerator Message Socket
AMI: Accelerator Message Interface
ASN: Accelerator Session

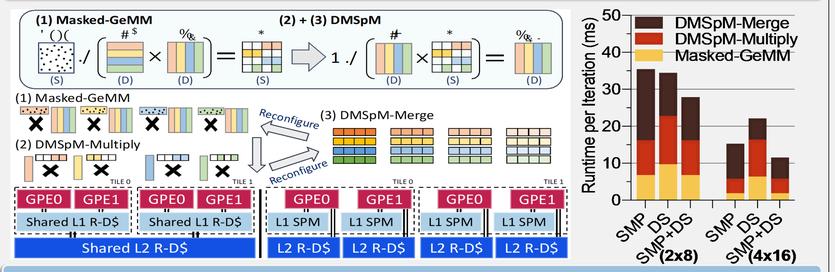
API Call	Parameters	Return Val.
TMLaunch()		
TMConfig()	int, int	
TMInspectCounters()	char *	double
TMReconfig()	config_e	bool
TMCompileKernel()	char *	bool
TMCompileSched()	char *	bool

- MSG_SEND/ MSG_SEND_ACK
- CRED_REQ/ ACK
- CRED_RETURN/CRED_RETURN_ACK
- CRED_RECALL/CRED_RECALL_ACK
- CRED_SEND/CRED_SEND_ACK

Algorithm Mapping

Category	Kernels	Configurations
Matrix Multiplication	Dense Matrix-Matrix Multiplication	Shared Cache / Systolic Array
	Sparse Matrix-Matrix Multiplication	Private SPM (L1), Private Cache (L2) / Systolic Array
Graph Processing	Breadth First Search, Page Rank	Shared Cache-Scratchpad (hybrid)
Linear Algebra	QR Decomposition	Systolic Array
	Triangular Matrix Solver	Shared Cache
Machine Learning	Neural Networks, Stochastic Gradient Descent	Shared Cache-Scratchpad (hybrid)
	Matrix Completion	Private SPM (L1), Shared Cache (L2)
	Sinkhorn	Shared Cache / Private SPM (L1), Private Cache (L2)
Signal Processing	Fast Fourier Transform, Mel-Frequency Cepstrum Coefficient	Systolic Array

Example Mapping: Sinkhorn Distance



- **SMP:** shared cache (L1/L2); **DS:** private SPM (L1) / private cache (L2)
- **SMP+DS:** Masked-GeMM and DMSpM-Multiply in **SMP** / DMSpM-Merge in **DS**
- **136.5x** and **31.3x** better EDP compared to an Intel Skylake CPU and NVIDIA Volta GPU

Notable Accomplishments

Software

- Integrated programming stack from user-level Python to Transmuter code
- Coverage metric to guide the programmer towards achieving optimal performance
- Automatic generation of optimized code from naïve serial C++ code for dense kernels

Hardware

- Profiling and analysis of 24 workflows (16 assigned + 8 non-assigned)
- 40 nm sparse matrix multiplier chip (predecessor to Transmuter) accepted for VLSI Symposium
- Timing simulator supporting multiple configs & functional simulator for rapid software prototyping
- Efficient implementations of prominent kernels - FFT, GeMM, GeMV, SpGeMM, MFCC, *etc.*

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.

THE ELECTRONICS RESURGENCE INITIATIVE