Accelerating Software 2.0
Foundations for Next-Generation Computer Systems

Christopher Aberger
Director of Software Engineering
Software 1.0 vs Software 2.0

- Written in code (C++, ...)
- Requires domain expertise
  - Decompose the problem
  - Design algorithms
  - Compose into a system
- Programmer input: training data
- Written in the weights of a neural network model by optimization
- Reduced lines of code

Andrej Karpathy. Scaled ML 2018 talk
Software 2.0 is Dataflow

1000x Productivity
Google shrinks language translation code from 500k imperative LoC to 500 lines of dataflow (TensorFlow)
The Case for Learned Index Structures

Tim Kraska*  
MIT  
Cambridge, MA  
kraska@mit.edu

Alex Beutel  
UC Berkeley  
beutel@berkeley.edu

Ed H. Chi  
Stanford University  
chi@stanford.edu

Jeffrey Dean  
Google, Inc  
d dean@google.com

Christopher Ré*  
University of British Columbia  
chris.re@ubc.ca

HoloClean: Holistic Data Repairs with Probabilistic Inference

Snorkel: Rapid Training Data Creation with Weak Supervision

Alexander Ratner  
Stanford University  
aratner@cs.stanford.edu

Stephen H. Bach  
Stanford University  
bach@cs.stanford.edu

Henry Ehrenberg  
Stanford University  
ehrenberg@cs.stanford.edu

Jason Fries  
Stanford University  
jfries@cs.stanford.edu

Sen Wu  
Stanford University  
senw@cs.stanford.edu

Christopher Ré  
Stanford University  
chris.re@stanford.edu

{jratner, bach, henryre, jfries, senwu, chrismre}@cs.stanford.edu
Next gen Software 2.0 systems need support for:

- **Hierarchical parallel pattern Dataflow**
  - Natural ML execution model

- **Terabyte sized models**
  - Higher accuracy

- **Sparsity**
  - Graph based neural networks

- **Flexible mapping**
  - Model and data parallelism

- **Data processing**
  - SQL in inner loop of ML training
Too Hot

Goldilocks Zone

Too Cold

©2020 SambaNova Systems Confidential | Do Not Distribute
Yesterday’s Goldilocks Zone is Constraining Progress

Faster, Higher Quality on Today’s models—better on Tomorrow’s models
How do we break out of the Goldilocks Zone?

Fundamental advances required at all layers of the stack.
The SambaNova Systems Advantage: Reconfigurable Dataflow Architecture

Full stack co-engineering yields optimizations where best delivered with the highest impact
SambaFlow Open Software for DataScale Systems

Graph Entry Points
- Write to OSS ML frameworks or user’s graph
- Push-button automation path

API Entry Point
- User programs to DSL
- Mix of manual and automatic

Optimizations
- Model parallel
- Data parallel
- Tiling
- Streaming
- Nested pipelining
- Op parallel
The Chip
• First Reconfigurable Dataflow Unit (RDU)
• TSMC 7nm
• 40B transistors
• 50 Km of wire
• 100s of TFLOPS
• 100s MB on chip
• Direct interfaces to TBs off chip

The System
Open standard rack,
Open standard form factor,
Open standard power,
Open standard cooling,
Open standard operations …
Reconfigurable Dataflow Unit (RDU)

Parallel Patterns

- map
- filter
- reduce
- groupBy

Array of reconfigurable compute, memory and communication

©2020 SambaNova Systems Confidential | Do Not Distribute
Spatial Dataflow Within an RDU

The old way: kernel-by-kernel

The Dataflow way: spatial

SambaFlow eliminates overhead and maximizes utilization
Rapid Dataflow Compilation to RDU

Weight
Input Data

Conv
Pool
Conv
Norm
Sum

Conv
Pool
Norm
Sum
SambaFlow Produces Highly Optimized Spatial Mappings

Dataflow Graph

Communication Pattern

SambaFlow Spatial Compilation

©2020 SambaNova Systems Confidential | Do Not Distribute
Uncompromised Programmability and Efficiency
Breaking out of the programmability vs. efficiency tradeoff curve

Energy Efficiency (MOPS/mW)

Reprogramming Time (seconds)

ASIC
Fixed-function
(Not reprogrammable)

FPGA
Fine-reconfigurable

RDU
Dataflow-based

GPU
Instruction-based

CPU
Instruction-based

©2020 SambaNova Systems Confidential | Do Not Distribute
The SambaNova Systems Advantage: Reconfigurable Dataflow Architecture

Full stack co-engineering yields optimizations where best delivered with the highest impact.
Model (Pipeline) Parallelism: Are we there yet?

1. Course Grained
2. HW Cost

Goal: Fine-Grained Model Parallelism
HW Cost: GPipe

**Panic:** Sacrifices latency for synchronous execution!
HW Cost: PipeDream
HW Cost: PipeDream
HW Cost: PipeDream
HW Cost: PipeDream
HW Cost: PipeDream
HW Cost: PipeDream
HW Cost: PipeDream
HW Cost: PipeDream

Panic: Sacrifices memory for synchronous execution!
Ideal Pipeline Parallelism Steady State

Goal: No hardware sacrifices!

Panic: Introduces asynchrony (delays).
Houston, we have a problem.

Key Insight: Scale your learning rate proportional to the delay.

\[ \alpha = \min \left( \alpha_{\text{sync}}, \frac{C}{T_i} \right) \]

Chris De Sa
Enabling Peak Dataflow Efficiency

PipeMare: Arxiv '20

**ResNet 50: Cifar 10**

Test Accuracy

- Epoch
- sync
- sn

**Transformer: IWSLT**

Test Loss

- Epoch
- sync
- sn

©2020 SambaNova Systems Confidential | Do Not Distribute
The SambaNova Systems Advantage: Reconfigurable Dataflow Architecture

Full stack co-engineering yields optimizations where best delivered with the highest impact.
How do we future proof our code?
What are the future models?

Models are the new code.
Enabling New Capabilities (0 ⇒ 1)

Trillion parameter NLP models
Key to knowledge understanding

High Resolution Deep Learning
50k x 50k
Astronomy, medical imaging, X-ray imaging, ...

Recommendation models with huge 100GB embedding tables
Recommendation is the backbone of internet services
Part 1: NLP

Models are the new code.
Proliferation of NLP Models

CreateML
FastBERT
Elmo, RoBERTa

Google
Microsoft
OpenAI
salesforce

BERT, XLNet
MT-DNN
Zero
GPT, GPT2
CTRL
Richer Context, In a Small Amount of Space

Microsoft open sources breakthrough optimizations for transformer inference on GPU and CPU

January 21, 2020

EMMA NING
Senior Program Manager, Azure Machine Learning

A three-layer BERT model in production at Bing.

Richer context, same space.
Richer, Contextual Information

3-wide encoders

24-slim encoders

More than 6x faster on Deeper BERT

Fewer Parameters, Better Quality on **Natural Language Inference**

QNLI: 3-layer 78.7 vs. Deeper 79

SambaNova enables Deeper Design Points
ZeRO & DeepSpeed: New system optimizations enable training models with over 100 billion parameters

February 13, 2020 | By DeepSpeed Team; Rangan Majumder: Junhua Wang

DeepSpeed + ZeRO

- Memory usage without ZeRO
  - Data
  - Model
  - ZeRO
  - Optimized memory usage

- With ZeRO
  - Data
  - Model
  - ZeRO
  - Optimized memory usage

Scale
- 10G parameter
- 10X bigger

Speed
- Up to 5X faster

Cost
- Up to 5X cheaper

Usability
- Minimal code change

The new language model our teams built is the largest and most powerful one ever created – a milestone with the promise to transform how technology understands and assists us.

Turing-NLG: A 17-billion-parameter language model by Mic...
This figure was adapted from a similar image published in DistillBERT. Turing Natural Language Generation (T-NLG) is...

©2020 SambaNova Systems Confidential | Do Not Distribute
Enabling Large Model Architectures With a Single System

Order of magnitude performance improvement, an order of magnitude fewer systems

“One Model” 1Trillion Params in a Single System: Same Programming Model
Models are the new code.

Part 2: Vision
Fast Growing Scale of Model Training Data
Evolution of high-resolution Deep Learning

- **Low-resolution** (e.g. cats)
- **4k images** (e.g. Autonomous driving)
- **50k x 50k** (e.g. astronomy, medical imaging, virus, ...)

©2020 SambaNova Systems Confidential | Do Not Distribute
Mapping High-Res Images to SambaNova

40k x 40k image running forward pass on UNet (image segmentation model)

SambaFlow automatically tiles the input image for deep learning operations and handles overlaps between tiles.

Tiles are streamed through model pipeline on chip.

- 3 x 40960 x 40960 input
- 409600 tiles per surface, or up to 26 million tiles for 64 channels
- GPU fails to allocate memory
- Even CPU errors out in PyTorch!

Only SambaNova can run these workloads out-of-the-box

©2020 SambaNova Systems Confidential | Do Not Distribute
No Compromise High-Res

**Classic tiling:**
chop image into sub-images

**SN tiling:**
handles overlaps across tiles based on network

**Identical result as non-tiled!**
And that’s just the tip of the iceberg...

GANs, Reinforcement Learning, Time Series, GCNs, PCA, and many more.
SambaNova: Breaking the Goldilocks Barriers, for Everyone

Performance Advantage

Highly Detailed Models  GPU Optimized Models  Bigger Models

Performance

GPU Optimized Models
Reconfigurable Dataflow for Unprecedented Flexibility

Performance balances computation & communication

Bottleneck: Yesterday’s platforms only program compute

Flexibility unlocks:
• 10x performance
• 0-to-1 applications

We’re hiring: sambanova.ai