

People

Mitchell Nahmias - Sphere Semi
Joshua Marks - Syntiant
Seth Morton - Extropic
Anton Sugolov - Extropic
Alexander Neagoe - Extropic
Daria Soboleva - Cerebras
Andy Zhang
Sung Jae
Armins Stepanjans
Michael Keating

Introduction

[- Sutro Group: top level

<https://github.com/cybertronai/gradient-checkpointing>

How to measure energy of an algorithm?

[- sutro follow-up

Spatial computer
Bill Dally paper

Overview of last 4 weeks

Trying to get away from manual memory placement: (through proxy metric, ByteDMD)
history of ByteDMD
Tiled matrix multiplication (cache efficiency)

https://github.com/cybertronai/ByteDMD/tree/dev/experiments/tiled_matmul
working set size

- smaller caches have smaller energy costs, hence rank algorithms by cache friendliness

[visualizations](#)

https://yaroslavvb.github.io/ByteDMD-vis/dotproduct_stack-mattson-linear.html

Tiled matrix multiplication is cache-friendly

But, looking at [grid](#) big gap for byteDMD
problem: temporaries

liveByteDMD

https://yaroslavvb.github.io/ByteDMD-vis/dotproduct_stack-live-linear.html

Manhattan distance [link](#)

Want to go back to Bill Dally model

lower bound from Tarjan

upper bound from Log-Structured Merge trees ([doc](#))

[gmail](#)

proof from [gemini](#)

(couple of steps were missing. Impossible to fill in).

-

Seth, Yad, Sung Jae looked at better implementations

<https://github.com/cybertronai/SutroYaro>

<https://github.com/cybertronai/SutroYaro/pulls>

Yaroslav looked at proxy metrics ([grid](#)) Three useful metrics:

- lpad (global density, local density, polymatroid)

Lessons

Visualize everything: <https://gf-2-sparse-parity-solver-400699997518.us-west1.run.app/>

Factor out tasks into scripts:

Instead of manually generating, create a script to generate and run a script

Agents will find holes in your metric:

0. Python escape mechanisms

1. If you fix input, they optimize to the input

2. Introduce timeouts

Use bounds to find bugs.

- Implement bound (lowest cost possible)

- Implement 45 algorithms

- Ask to optimize them

- Once it's below bound -- ask to find bugs (either in alg or in bound)

Organizing personal memory:

<https://github.com/cybertronai/ByteDMD/tree/dev/gemini>

Future tasks

[- Yaroslav's sutro planning sprint #1

Past:

([dev](#) branch)

Future:

<https://github.com/cybertronai/ByteDMD/blob/dev/docs/simplified-explicit-communication-model.md>

1. Implement sparse parity on a 2D grid

2. Turn this into a challenge

Justification: Bill Dally's proposed model of computation [acm](#).

- Teaches at Stanford (ie, green electronics [class](#))

- Chief Scientist of Nvidia, joined in 2009

three-address code (TAC). It's the classic compiler IR shape — each instruction is `op dst, src1, src2` (or `op src1, src2, dst` depending on convention) with at most three operands, all referring to addressable storage. LLVM IR is TAC, GCC's GIMPLE is TAC, most teaching compilers use TAC.

13,21,1,4,2,6

add 15, 23, 3

copy 23, 52

2,4,6,19

Links

<https://cybertronai.github.io/ByteDMD/grid>

source: <https://github.com/cybertronai/ByteDMD/tree/dev/experiments/grid>

[visualizations](#)

<https://github.com/cybertronai/ByteDMD/tree/dev/gemini#april-20-2026>

python instructions: ([chagpt](#))

Ideas for IR:

<https://claude.ai/chat/5e4b13eb-a1bb-44de-b0e9-eae90b39d9eb>

<https://claude.ai/chat/7f9973ff-d95f-4906-a234-7da8a850f3cc>