PAUL SCHERRER INSTITUT

PSI

Filip Leonarski :: Beamlines Data Scientist :: MX Data Group

Addressing protein serial crystallography 36 GB/s data-rate challenge with FPGAs and GPUs

ICALEPCS 2023: 3rd Data Science and Machine Learning Workshop

Cape Town, October 7th, 2023

Two X-ray facilities: Swiss Light Source synchrotron and SwissFEL Free Electron Laser

Kilohertz Serial Synchrotron Crystallography at SLS 2.0

- **Serial crystallography** solves protein structures with X-ray diffraction images from thousands of crystals

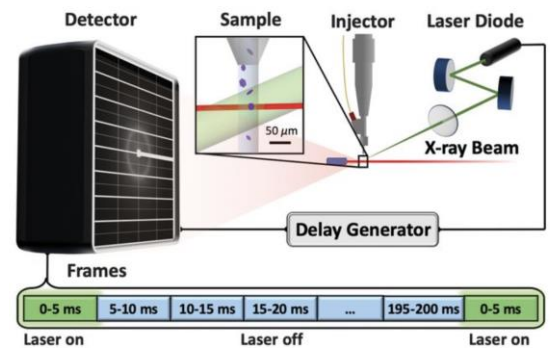
- Allows to visualize how proteins interact at milli- or microsecond time scales

- **Very data intensive technique**
  - Usually 5-10 time points are of interest
  - 50k images needed to solve a structure at one time point
  - Most (90%) images don’t capture protein crystals

  - Easily need 5 million diffraction images
  - Each image is 18 MB

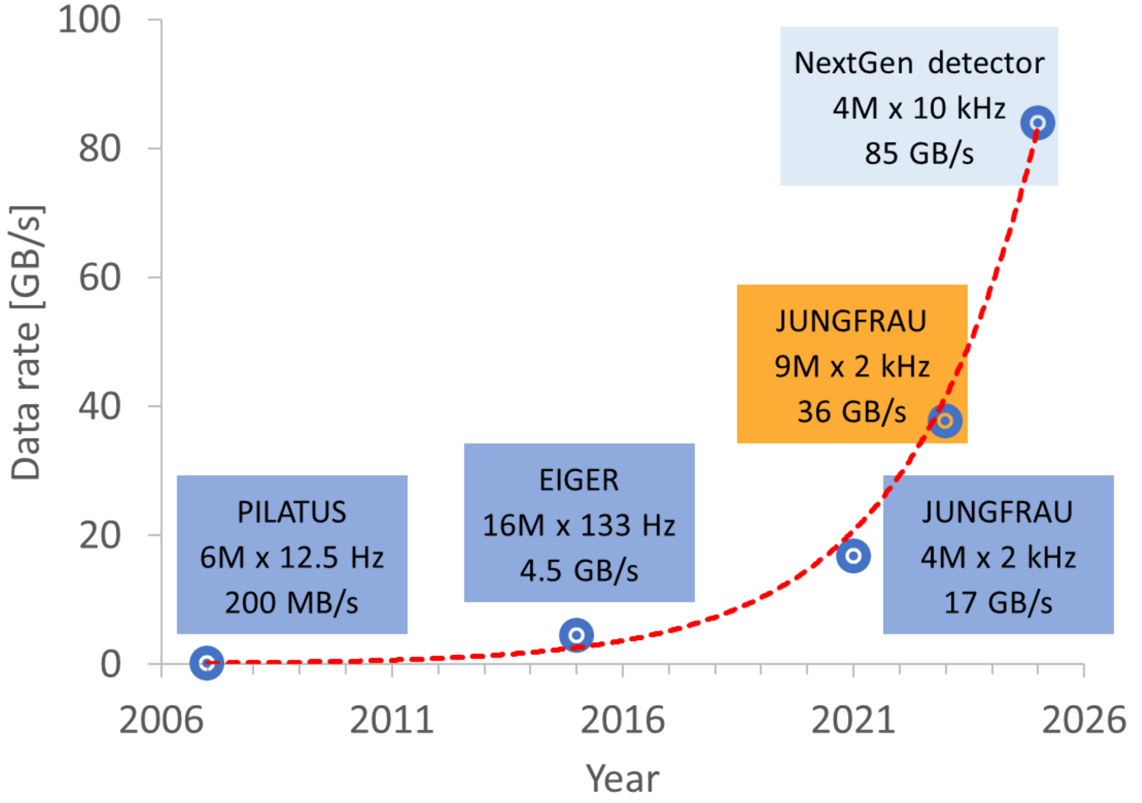
  - **100 TB per experiment**

- Challenging for the IT infrastructure
  - Acquisition
  - Storage
  - Analysis



T. Weinert et al., *Science* (2019)

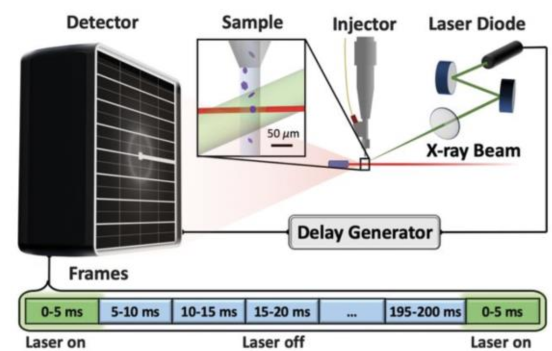
Better detectors = better science = bigger challenge for IT



F. Leonarski et al., *J Synchrotron Rad.* (2022)

Kilohertz Serial Synchrotron Crystallography at SLS 2.0

- **Serial crystallography** solves protein structures with X-ray diffraction images from thousands of crystals
- Allows to visualize how proteins interact at milli- or microsecond time scales
- **Very data intensive technique**
  - Usually 5-10 time points are of interest
  - 50k images needed to solve a structure at one time point
  - Most (90%) images don't capture protein crystals
  - Easily need 5 million diffraction images
  - Each image is 18 MB
  - **100 TB per experiment**
- Challenging for the IT infrastructure
  - Acquisition
  - Storage
  - Analysis



T. Weinert et al., *Science* (2019)

More details – the very last mini-oral presentation on Thursday

Kilohertz Serial Synchrotron Crystallography at SLS 2.0

- **Serial crystallography** solves protein structures with X-ray diffraction images from thousands of crystals

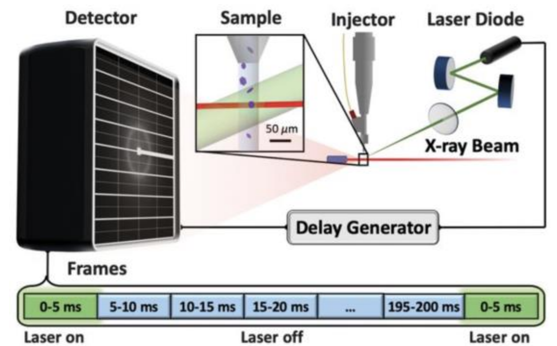
- Allows to visualize how proteins interact at milli- or microsecond time scales

- **Very data intensive technique**
  - Usually 5-10 time points are of interest
  - 50k images needed to solve a structure at one time point
  - **Most (90%) images don't capture protein crystals**

  - Easily need 5 million diffraction images
  - Each image is 18 MB

  - **100 TB per experiment**

- Challenging for the IT infrastructure
  - Acquisition
  - Storage
  - Analysis

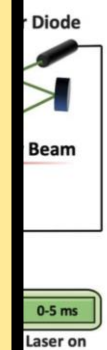


T. Weinert et al., *Science* (2019)

Kilohertz Serial Synchrotron Crystallography at SLS 2.0

- **Serial crystallography** solves protein structures with X-ray diffraction images from thousands of crystals
- Allows to visualize how proteins interact at milli- or
  n
- **V**

Significant reduction is possible if one could find frames that contain information content and suppress saving all the others



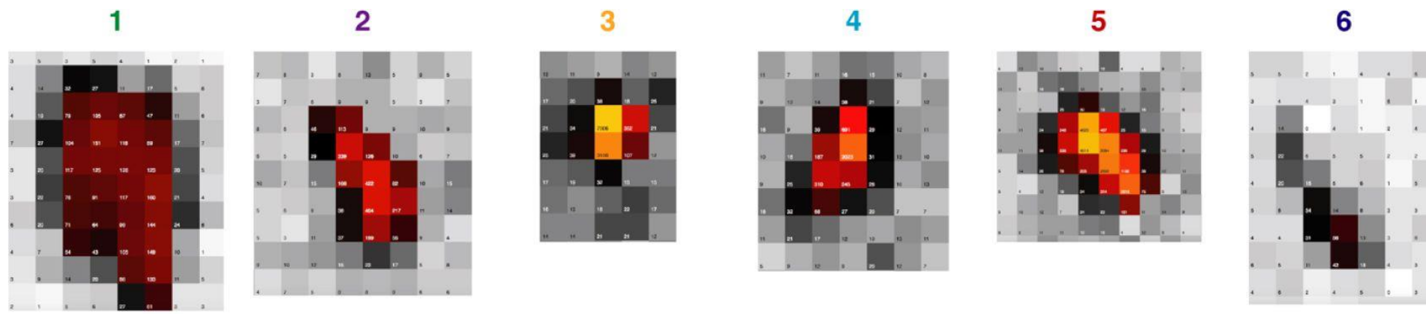
9)
aw8634

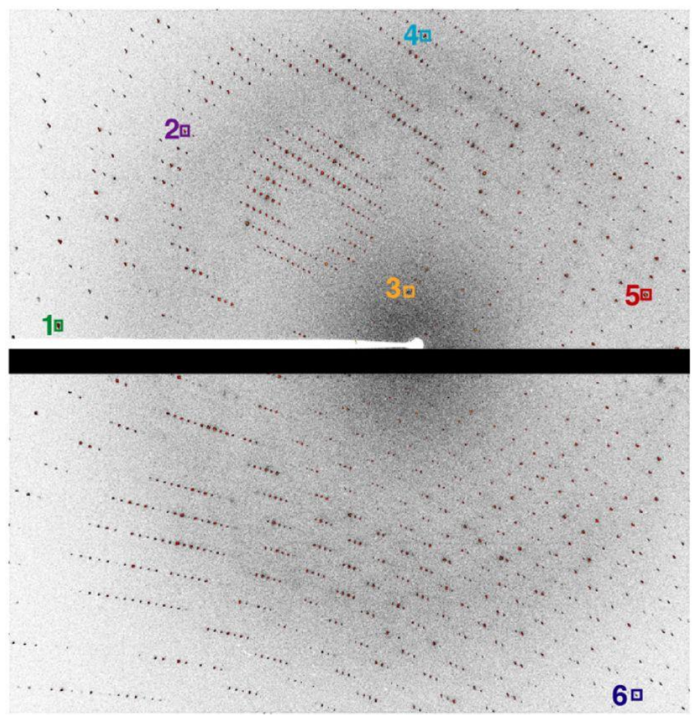
  - **100 TB per experiment**

- Challenging for the IT infrastructure
  - Acquisition
  - Storage
  - Analysis

Two steps needed: spot finding and indexing

Usual protein diffraction image





Spot finding

Generate list of spot centroids from the image

=> Local
=> Number of spots indicates if crystal was encountered

Indexing

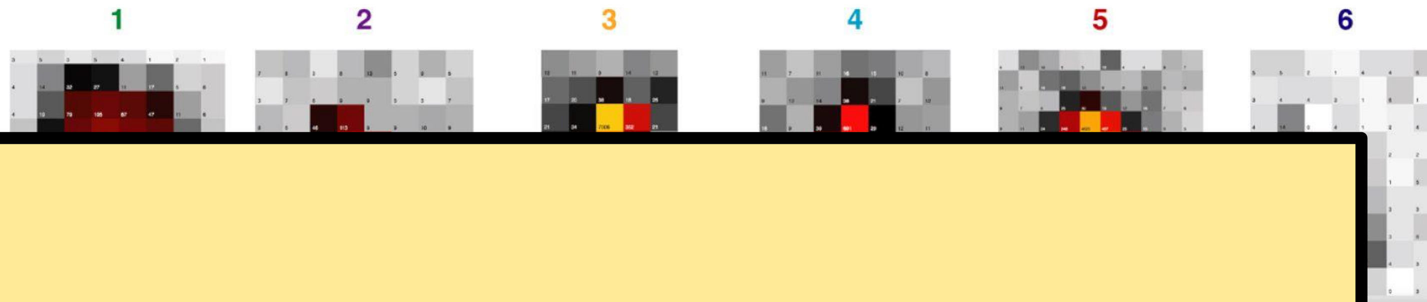
Fit spot positions into a crystal lattice

=> Whole image
=> Failure means diffraction doesn't come from a crystal

F. Leonarski et al., *Nat Methods* (2018)

Two steps needed: spot finding and indexing

Usual protein diffraction image



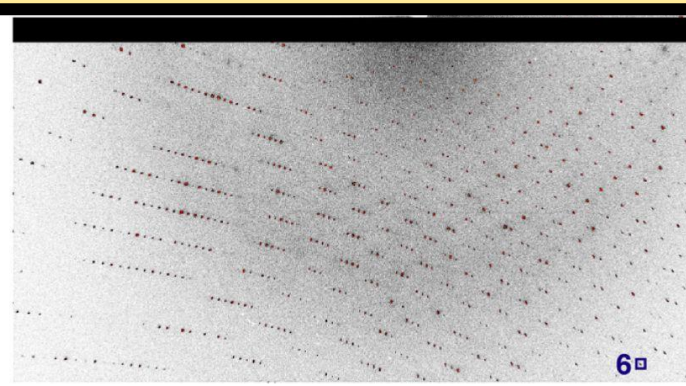
Both problems have known algorithms to solve them

However current CPU implementations are too slow

Spot findin

Generate li
centroids from the image

=> Local
=> Number of spots indicates if crystal was encountered



ions into
a crystal lattice

=> Whole image
=> Failure means diffraction doesn't come from a crystal

F. Leonarski et al., *Nat Methods* (2018)

PAUL SCHERRER INSTITUT

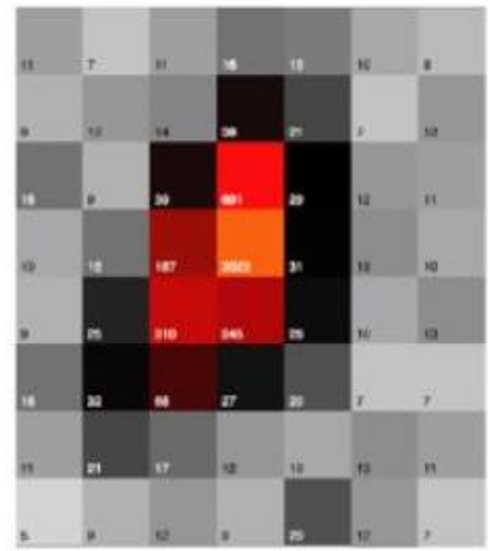
Spot finding

Spot finding

- Currently – using deterministic algorithms:
  - Threshold based on **pixel intensity**
  - Threshold based on **mean and variance of the pixel neighborhood**
  - “connect” pixels above threshold (DBSCAN-like)

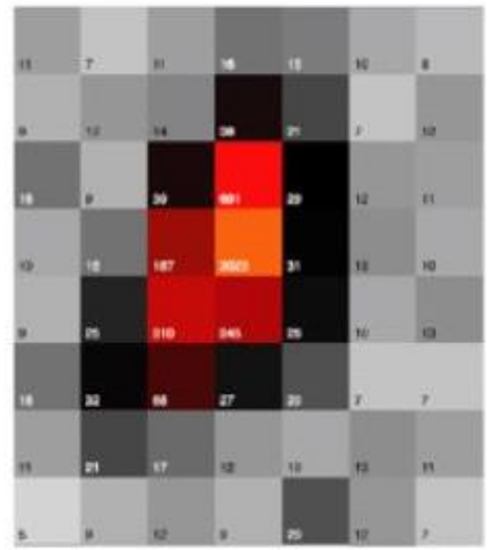
- Main limitation: **hyperparameters** selection

- Looking into **convolutional neural networks**
  - Few interesting approaches (Stanford, Berkeley), but require retraining model for a particular crystal type
- **Reinforcement learning** to learn hyperparameters (EuXFEL)



Deterministic spot finding

- Need to quickly calculate mean and standard deviation of 11x11 size boxes around each pixel
- We have done **three** implementations:
  – CPU
  – GPU
  – FPGA
- **CPU implementation is slower** that what needed – used only as reference
- **GPU and FPGA are fast** (limited by PCIe bandwidth), but each implementation comes with own limitations



PAUL SCHERRER INSTITUT

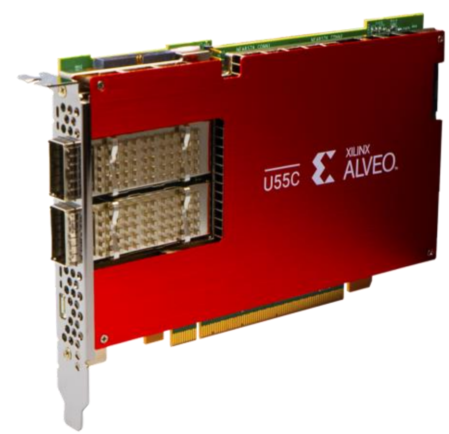
GPU vs. FPGA

GPU

- Similar to CPU
- Software development
- Parallel architecture with multiple streaming multiprocessors
- Single instruction multiple data model
- Both floating-point and integer math

FPGA

- Design yourself electronics
- Hardware design
- Parallelism via assigning functions to different chip areas
- Access to network
- Integer arithmetic preferred



GPU vs. FPGA

GPU

- 2 weeks of work
- C++ dialect (CUDA)
  - Easier to program for SW developer
- Easy to test on laptop
- Performance is non-deterministic
  - Multiple kernels might be running; Nvidia driver is scheduling execution
- Images need to be loaded from memory
- Competition with other GPU tasks

FPGA

- 3 months of work
- C++ dialect (Xilinx HLS)
  - Steep learning curve
- Lot of effort to test
- Performance is deterministic
  - Performance can be predicted beforehand
- Images can be loaded from network (=> we use FPGAs as network cards)
- Full parallelism => no competition

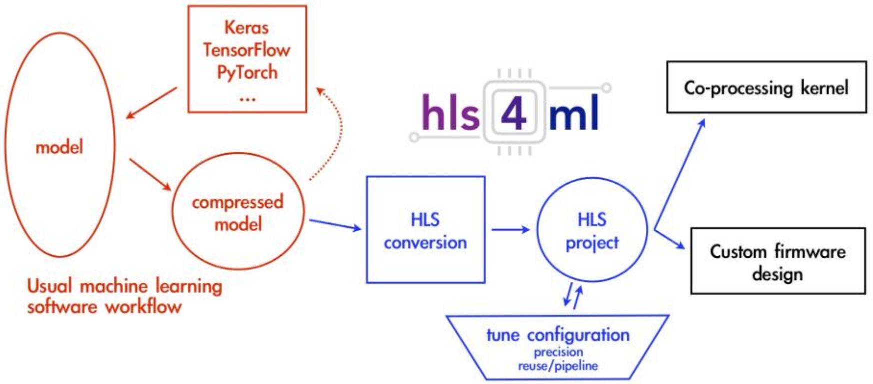
Looking forward for ML spot finding

- Why? **Hyperparameters of deterministic algorithms**

- **GPUs** are natural for machine learning, all the tools are available, e.g. with PyTorch

- **ML frameworks** don't support FPGA out-of-the-box
  - hls4ml framework from CERN
  - Vitis AI from Xilinx (currently AMD)
  - Mipsology Zebra (currently AMD)

  - **hls4ml** is the easiest to interface into existing FPGA designs



PAUL SCHERRER INSTITUT

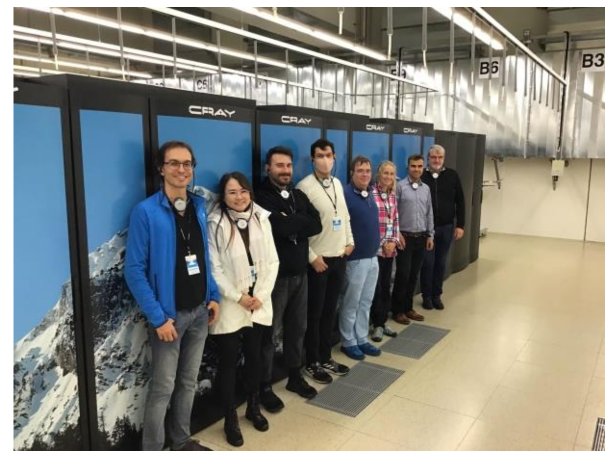
Indexing

Reduction of High Volume Experimental Data using Machine Learning (RED-ML)

- Broad collaboration:
  - Swiss Data Science Center
  - Swiss National Supercomputing Centre
  - PSI (Science IT and MX)

- Swiss Data Science Center supports Swiss researchers with data science and ML expertise

- Support of SDSC employees is provided through grant application system



SDSC

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Indexing

- Indexing is linear algebra optimization problem
  - ML solution is unlikely

Indexing

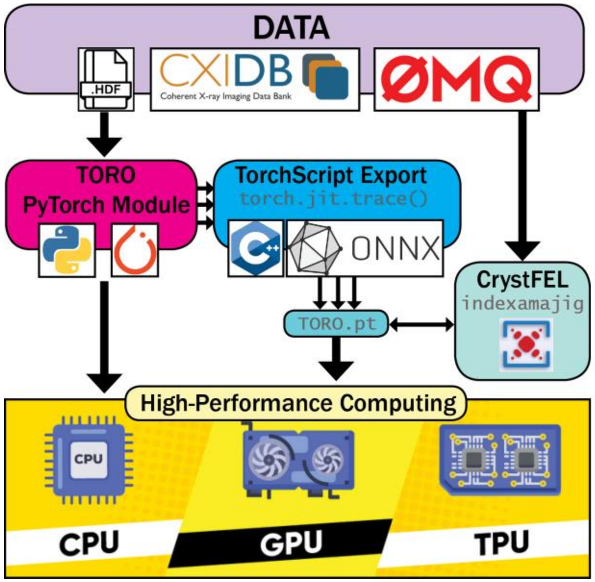
- Indexing is linear algebra optimization problem
  - ML solution is unlikely

- However – **ML tools** are handy!

Indexing

PyTorch

- Indexing is linear algebra optimization problem
  – ML solution is unlikely
- However – **ML tools** are handy!
- We have implemented an indexing in **PyTorch**
- Code developed in Python, but compiled on-the-fly to low-level representation
- The same code can be deployed on **CPU and GPU** (and even just for fun on TPU)
- Significantly **shorter code** than low level implementation
- Performance is **very high**
  – Though "pure" CUDA is faster



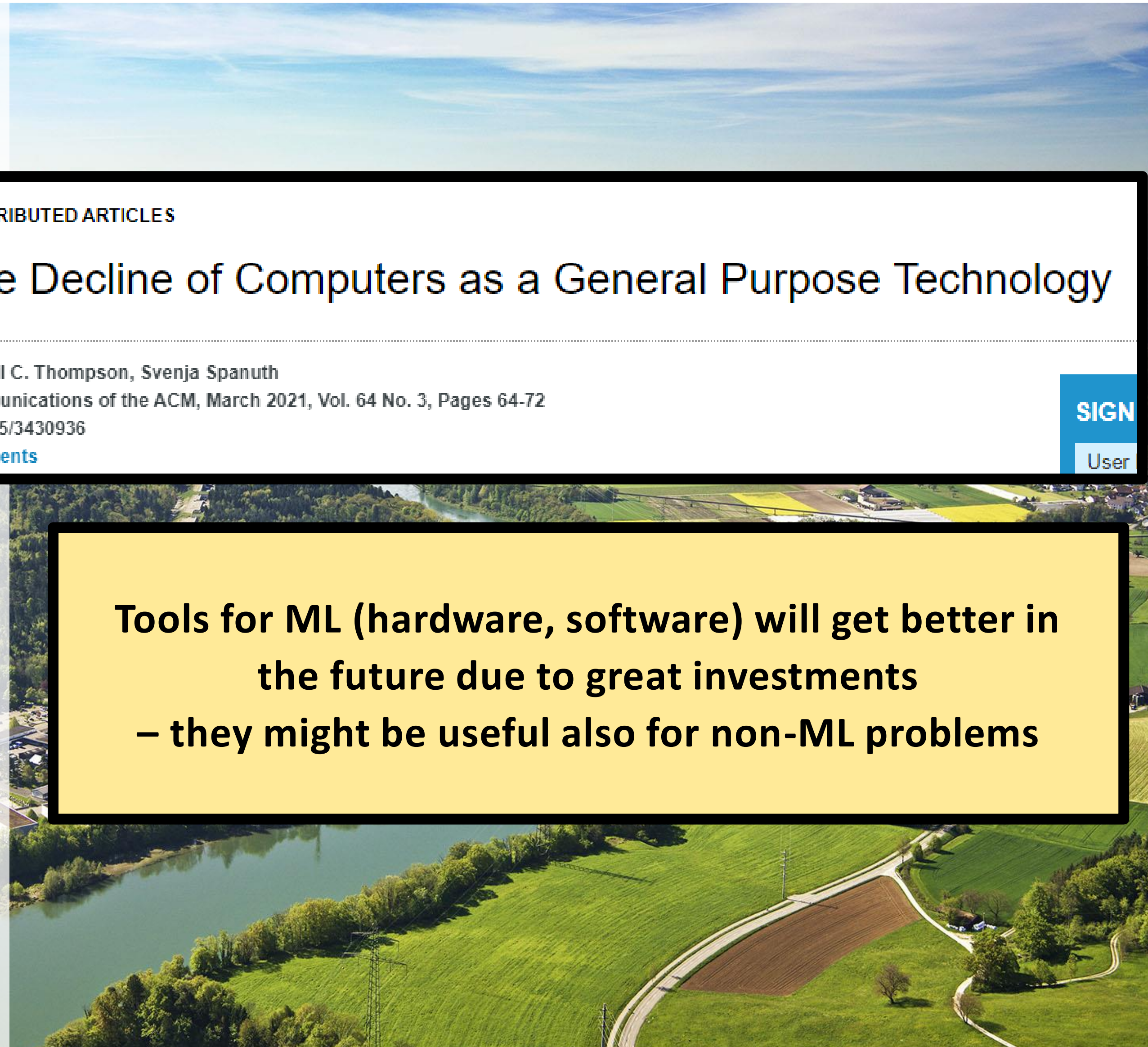
Courtesy of P. Gasparotto, L. Barba and H.-C. Stadler

Conclusions

- When handling high data throughput GPUs and FPGA are helpful
- GPUs have lower entry barrier and are preffered for floating point calculations
- FPGAs have higher development cost, but have extra benefits (e.g., network)
- Machine learning frameworks can be used for non-ML problems

Conclusions

- When handling high data throughput GPUs and FPGA are helpf
- GPUs have lower e barrier and are preffered for floati point calculations
- FPGAs have higher development cost, but have extra benefits (e.g., network)
- Machine learning frameworks can be used for non-ML problems



CONTRIBUTED ARTICLES

The Decline of Computers as a General Purpose Technology

By Neil C. Thompson, Svenja Spanuth
Communications of the ACM, March 2021, Vol. 64 No. 3, Pages 64-72
10.1145/3430936
Comments

SIGN
User

Tools for ML (hardware, software) will get better in the future due to great investments
– they might be useful also for non-ML problems

Acknowledgements

- **MX Group (PSI)**
  - Meitian Wang
- **Science IT (PSI)**
  - Alun Ashton
  - Piero Gasparotto
  - Markus Janousch
  - Hans-Christian Stadler
- **Swiss Data Science Center**
  - Luis Barba
  - Benjamin Béjar
- **Swiss National Supercomputing Center**
  - Henrique Mendoça
- **MAX IV**
  - Jie Nan
  - Zdeněk Matěj
- **Diamond Light Source**
  - Nicolas Devenish
  - Richard Gildea
  - Graeme Winter

Contact information

filip.leonarski@psi.ch