

Tematy prac mgr  
Michał Hanćkowiak

[mhanckow@amu.edu.pl](mailto:mhanckow@amu.edu.pl)

b3-34, wtorek 12:00-13:00,  
środa 15:30-16:30

# Sprzętowe przyspieszanie uczenia sieci neuronowych

Zastosowanie GPU (zamiast CPU) do obliczeń występujących przy uczeniu sieci neuronowych

Także inny sprzęt: układy FPGA, Tensor Processing Unit, itp  
Wykorzystanie bibliotek OpenCL lub OpenGL lub innych do przyspieszania obliczeń...  
jak to się robi przy pomocy powyższych bibliotek...

## **Projekt b-r:**

przyspieszenie biblioteki FANN przy pomocy biblioteki OpenCL

<https://github.com/libfann/fann>

<https://www.khronos.org/opencl/>

FANN zawiera backprop i jego bardziej wyrafinowane wersje Rprop Quickprop

Przetestować jak to się sprawuje na różnych kartach graf !!!

# Linux vs sieci komputerowe, podsys „iproute2”, dyscypliny kolejkowe = qdisc, „kształtowanie ruchu sieciowego”, Diffserv ?

Qdisc przypisana do if wstrzymuje/ zmienia kolejność pakietów na podstawie różnych kryteriów...

Należy omówić jak działają różne qdisc:

<https://linux.die.net/man/8/ip> <https://linux.die.net/man/8/tc>

<https://tldp.org/HOWTO/Adv-Routing-HOWTO/index.html>

<https://man7.org/linux/man-pages/man8/tc-sfq.8.html>

<https://man7.org/linux/man-pages/man8/tc-drr.8.html>

<https://man7.org/linux/man-pages/man8/tc-cbq.8.html> Jest w NS-2 i w Linuxie !!!

<https://man7.org/linux/man-pages/man8/tc-tbf.8.html>

<https://man7.org/linux/man-pages/man8/tc-htb.8.html> Tylko w Linuxie

<https://man7.org/linux/man-pages/man8/tc-red.8.html>

## Projekt b-r:

1. do symulatora NS-2 dodać te qdisc, które są w Linuxie, ale brakuje ich w NS-2 (np. HTB)

NS-2: <http://www.isi.edu/nsnam/ns>

[http://mhanckow.vm.wmi.amu.edu.pl:20002/zajecia/\\_xowiki2/SIK\\_cw#temG](http://mhanckow.vm.wmi.amu.edu.pl:20002/zajecia/_xowiki2/SIK_cw#temG)

2. stworzyć eksperyment na maszynach wirt (Vbox ?)

pokazujący efekty zastosowania qdisc...

# Linux vs sieci komputerowe, podsys „iproute2”, tunele, konfiguracja IPsec

Tworzenie różnych typów tuneli (np. GRE),  
konfiguracja IPsec (zabezpieczenia na poziomie warstwy 3),

<https://man7.org/linux/man-pages/man8/ip-xfrm.8.html> ipsec vs "ip xfrm"

<https://en.m.wikipedia.org/wiki/IPsec>

<https://ibug.io/blog/2021/10/linux-ipsec-with-ip-xfrm/>

<https://croz.net/news/xfrm-programming/>

<https://cromwell-intl.com/networking/what-is-ipsec.html>

<https://datatracker.ietf.org/doc/html/rfc4301> IPsec

<https://datatracker.ietf.org/doc/html/rfc4306> IKEv2

## **Projekt b-r:**

stworzyć eksperyment na maszynach wirt (Vbox ?)

pokazujący efekty zastosowania tuneli oraz konfiguracji IPsec

# Linux vs podsyst audio

## OSS, ALSA, PulseAudio, ...

Omówienie obsługi dźwięku w Linuxie:

OSS, ALSA, ser dźwiękowe jak PulseAudio,

Jak to działa? Jak się konfiguruje?

Co można zrobić oprócz tradycyjnych zastosowań ???

<https://www.volkerschatz.com/noise/alsa.html>

<http://delogics.blogspot.com/2014/11/understanding-alsa-device-subdevice-and.html>

<https://www.alsa-project.org/alsa-doc/alsa-lib/index.html>

<https://www.alsa-project.org/main/index.php/Asoundrc>

<https://ubuntu.pl/czytelnia/2012/02/23/obsługa-dźwięku-na-linuksie-cz-1-alsa/>

<https://www.kernel.org/doc/html/latest/sound/designs/oss-emulation.html> alsa vs oss

<https://github.com/i-rinat/apulse> apulse (emulacja pa pod alsa)

### **Projekt b-r:**

**sieć fizyczna wykorzystująca dźwięk zamiast fal em...**

**Co daje „rozpraszanie widma” na przykładzie fal**

**akustycznych, zast kodów wykr/popr błędy, transf**

**Fouriera itp. - na użytek zajęć typu SIK...**

# algorytmy rozproszone odporne na (złośliwe?) błędy w sieci, „fault tolerance”

Dolne oszacowania oraz alg rozwiązujące problemy takie jak:

## **Consensus**

„procesory” w sieci muszą się zgodzić na jedną liczbę,  
spośród podanych na we,  
procesory się psują w ograniczonym zakresie,  
a mimo to algorytm działa...

## **K-set-agreement**

uogólnienie problemu consensus... (k-wartości, zamiast 1)

Literatura:

[http://mhanckow.vm.wmi.amu.edu.pl:20002/zajecia/\\_xowiki2/ALR31S\\_wy](http://mhanckow.vm.wmi.amu.edu.pl:20002/zajecia/_xowiki2/ALR31S_wy)

Hagit Attiya "Distributed algorithms", rozdz 11, (książka)

Chaudhuri, Herlihy, Lynch, Tuttle, "Tight Bounds for k-Set Agreement"

Raczej dla osób które na „I poziomie” wybrały przedmiot ALR...

## **Projekt b-r:**

symulacja + wizualizacja powyższych algorytmów

# Przyspieszanie działania kodu w językach skryptowych

- Chodzi o sposoby używane w językach skryptowych/dynamicznych aby programy działały szybciej...
- Mowa o tłumaczeniu kodu do bajt-kodu, różnych optymalizacjach w bajt-kodzie oraz w maszynie wirtualnej, która go wykonuje...
- Także chodzi o pewne metody przyspieszające operacje na strukturach danych...  
(patrz: reprezentacja wewn str danych w j. Tcl)
- Można porównać 2 lub 3 języki skryptowe np. Tcl, Python, JavaScript, omówić i porównać zastosowane rozwiązania
- **Projekt:** eksperymenty z powyższymi mechanizmami; który język robi to lepiej ??