



Dmitry A. Zaitsev - profesor Wyższej Szkoły Informatyki i Zarządzania w Rzeszowie, senior member ACM i IEEE, ostatnio profesor wizytujący w Université Côte d'Azur, Francja. W 2017 roku był profesorem wizytującym w The University of Tennessee Knoxville, USA, na stypendium Fulbrighta, pracował w Innovative Computing Laboratory kierowanym przez Jacka Dongarrę. Efektem tego pobytu jest wspólnie opublikowana praca oraz wydane oprogramowanie ParAd.

## **Sleptsov Net Computing rozwiązuje problemy współczesnych superkomputerów. Wystąpienie dr Jacka Dongarry jako laureata nagrody Turinga w listopadzie 2022 roku**

[Wystąpienie laureata nagrody Turinga Jacka Dongarra](#) z 2021 roku na International Conference for High Performance Computing, Networking, Storage, and Analysis (SC22) rzuca nowe światło na problemy rozwoju współczesnych superkomputerów. Jego głęboka analiza wydajności superkomputerów implikuje wnioski, że architektura najpotężniejszych komputerów na świecie nie jest w pełni przystosowana do rozwiązywania praktycznych zadań w trudnych dziedzinach zastosowań. W tej sytuacji Sleptsov Net Computing jawi się jako obiecujące podejście mogące zapewnić przełom w kwestii wydajności współczesnych superkomputerów.

Koncepcja Sleptsov Net Computing (SNC) została opisana w moim artykule „Sleptsov Nets Run Fast” [1], a aspekty historyczne omówione zostały w znanym w USA publikatorze IGI-Global News Room w tekście [The Highest Standard: Sleptsov Software, on May 23, 2017](#). Artykuł „Strong Sleptsov nets are Turing complete” [2], (opublikowana przez prof. Zaitseva z afiliacją WSIZ w jednym z najlepszych światowych czasopism informatycznych wchodzących na listę CORE A – przyp. red. ) dowodzi z kolei matematycznie, że dowolny algorytm współbieżny może być określony przez sieć Sleptsova.

### **Wkład dr. Jacka Dongarry we współczesną informatykę i problemy poruszane w jego wystąpieniu**

Przez ponad dwie dekady wydajność superkomputerów była mierzona poprzez rozwiązywanie dużych gęstych losowych układów liniowych przy użyciu biblioteki [LINPACK](#). W swoim przemówieniu laureat nagrody Turinga dr Jack Dongarra nazywa LINPACK wskaźnikiem wysokiej wydajności obliczeniowej. W oparciu o ten wskaźnik powstała seria superkomputerów w USA i innych krajach, w tym najpotężniejszy obecnie komputer na świecie - Frontier z Oak Ridge Laboratory, USA. Przypomnijmy, że wydajność komputerów mierzy się w FLOPSach - operacjach zmiennoprzecinkowych na sekundę, a Frontier

przełamuje barierę exaflopsów, która oznacza moc dziesięć do osiemnastej potęgi. W końcowej części wykładu dr Jack Dongarra przedstawił uzyskane niedawno informacje o wydajności najlepszych komputerów, znajdujących się na liście [Top500](#), w rozwiązywaniu rzeczywistych zadań w różnorodnych dziedzinach zastosowań, reprezentowanych głównie przez symulatory prowadzące do rozwiązywania rzadkich układów liniowych. Frontier i wiele innych czołowych komputerów wykazuje wydajność rzędu 0,8%, co oznacza, że Frontier ma realną wydajność około dziesięciu petaflopów, a nie eksaflopów. Tylko japoński komputer Fugaku firmy Fujitsu, zajmujący ostatnio drugie miejsce, osiąga wydajność 3,0%.

Rank	Site	Computer	Country	Cores	Rmax [Pflop/s]	Top 500 Rank	HPCG [Pflop/s]	% of Peak
1	RIKEN Center for Computational Science	Fugaku, Fujitsu A64FX 48C 2.2 GHz, Tofu D, Fujitsu	Japan	7,630,848	422	2	16.0	3.0%
2	DOE / SC / ORNL	Frontier, HPE Cray Ex235a, AMD 3rd EPYC 64C 2 GHz, AMD Instinct MI250X, Slingshot 10	USA	8,730,112	1,102	1	14.1	0.8%
3	EuroHPC / CSC	LUMI, HPE Cray EX235a, AMD Zen 3 (Milan) 64C 2GHz, AMD MI250X, Slingshot-11	Finland	2,174,976	304	3	3.41	0.8%
4	DOE / SC / ORNL	Summit, AC922, IBM POWER9 22C 3.7 GHz, Dual-rail Mellanox FDR, NVIDIA Volta V100, IBM	USA	2,414,592	149	5	2.93	1.5%
5	EuroHPC / CINECA	Leonardo, BullSequana XH2000, Xeon Platinum 8358 32C 2.6 GHz, NVIDIA A100 SXM4 40 GB, Quad-rail NVIDIA HDR100 InfiniBand	Italy	1,463,616	175	4	2.57	1.0%
6	DOE / SC / LBNL	Perlmutter, HPE Cray EX235n, AMD EPYC 7763 64C 2.45 GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10	USA	761,856	70.9	8	1.91	2.0%
7	DOE / NNSA / LLNL	Sierra, S922LC, IBM POWER9 20C 3.1 GHz, Mellanox EDR, NVIDIA Volta V100, IBM	USA	1,572,480	94.6	6	1.80	1.4%
8	NVIDIA	Selene, DGX SuperPOD, AMD EPYC 7742 64C 2.25 GHz, Mellanox HDR, NVIDIA Ampere A100	USA	555,520	63.5	9	1.62	2.0%
9	Forschungszentrum Juelich (FZJ)	JUWELS Booster Module, Bull Sequana XH2000, AMD EPYC 7402 24C 2.8 GHz, Mellanox HDR InfiniBand, NVIDIA Ampere A100, Atos	Germany	449,280	44.1	12	1.28	1.8%
10	Saudi Aramco	Dammam-7, Cray CS-Storm, Xeon Gold 6252 20C 2.5 GHz, InfiniBand HDR 100, NVIDIA Volta V100, HPE	Saudi Arabia	672,520	22.4	20	0.88	1.6%

Wydajność współczesnych superkomputerów w rzeczywistych zadaniach na podstawie wykładu laureata nagrody Turinga dr Jacka Dongarry.

W istocie superkomputer nie jest tak prosty jak podział ziemi na działki w celu zmierzenia ich jedną miarą, ale wymaga złożonego, zrównoważonego podejścia polegającego na użyciu mieszanki symulatorów dla najważniejszych dziedzin zastosowań. Gęste systemy zapewniają regularne obciążenie procesorów i podsystemu komunikacyjnego dobrze wypełniającego bufor, aby wygładzić wąskie gardło procesor-pamięć. Zadania rzeczywiste, reprezentowane przez symulatory i systemy rzadkie, wymagają nieregularnego wzorca komunikacji, który jest dość dobrze implementowany przez sześciowymiarowy torus Fugaku Tofu D Interconnect. Tak więc projektowanie podsystemów komunikacyjnych w postaci wielowymiarowego torusa stanowi perspektywiczny kierunek badań, rozwijany przez prof. Zaitseva i jego współautorów z wykorzystaniem teorii nieskończonych sieci Petriego.



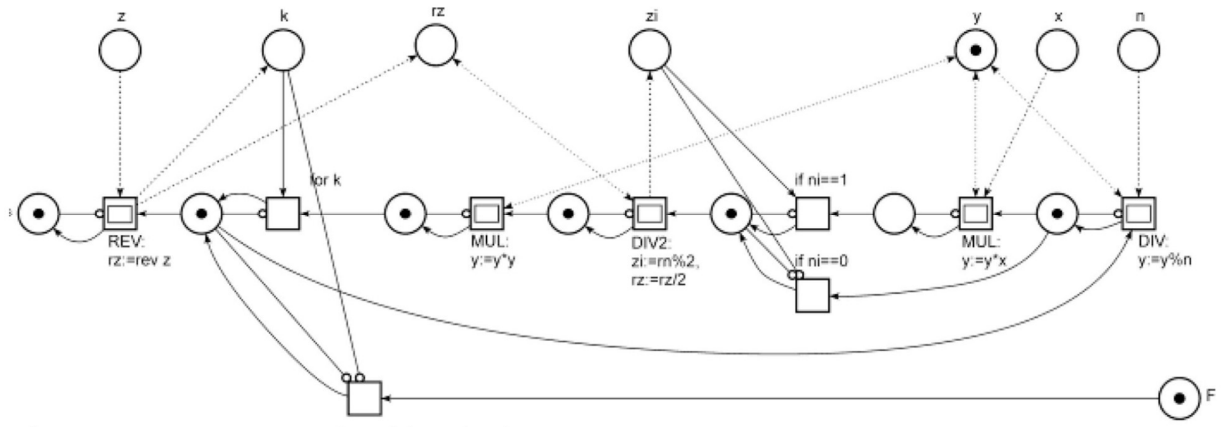
Superkomputer Fugaku

**Komputer oparty na sieci Sleptova nie wykazuje wąskiego gardła w układzie procesor-pamięć i wykazuje najwyższą wydajność**

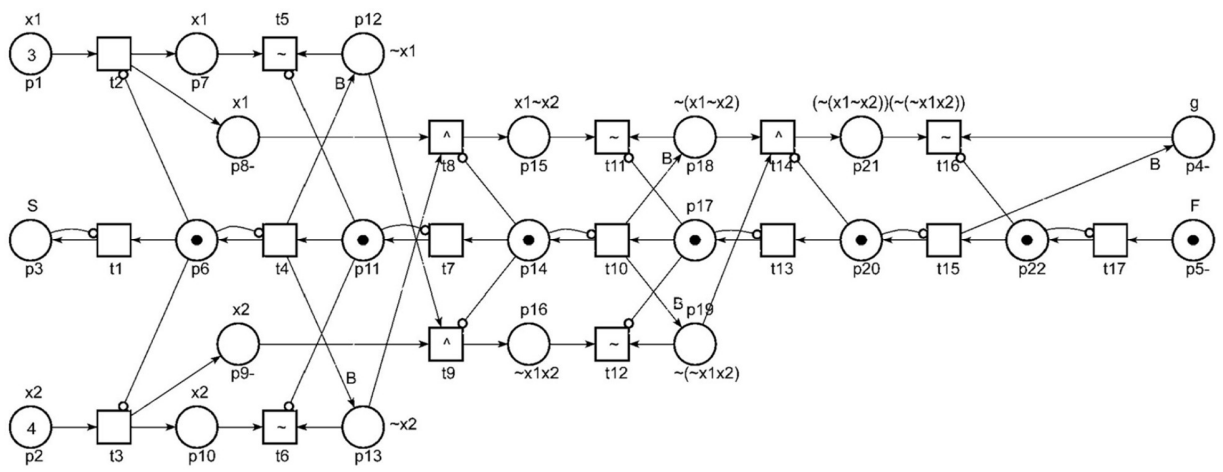
Sleptsov Net Computing idzie dalej; wykorzystuje pamięć obliczeniową, reprezentowaną jako dwu- i wielkoformatowe struktury, gdy uwzględnione są przestrzenne osobliwości modelowanych systemów. Wykonywanie obliczeń w pamięci pozwala nam pozbyć się tradycyjnego problemu wąskiego gardła procesor-pamięć. Zamiast wyrafinowanej, konwencjonalnej architektury do wykorzystania wielordzeniowych CPU i GPU w węzłach rozproszonych, na przykład z narzędziami zestawu OpenMP-CUDA-MPI, Sleptsov Net Computing oferuje jednorodną koncepcję zunifikowanego języka graficznego do programowania współbieżnego z bardzo drobną granulacją obliczeń równoległych. Jako powstała z teorii sieci miejsc i tranzycji, sieć Sleptsova odziedziczyła rozwinięte metody dowodzenia poprawności programów równoległych już na etapie wytwarzania oprogramowania. Implementacja procesorów sieci Sleptsova w postaci macierzy pamięci obliczeniowej prowadzi do nanosekundowych czasów obliczeń masywnie równoległych, co oznacza dobrą zdolność do sterowania obiektami hipersonicznymi w aplikacjach czasu rzeczywistego.

Osobnym zagadnieniem jest niezawodność oprogramowania. Wyobraźmy sobie, że niezawodność pewnego zbioru oprogramowania działającego w świecie opiera się na przekonaniu, że nie zawiera ono błędów lub raczej zawiera błąd z małym prawdopodobieństwem. Przekonanie to jest poparte pewną liczbą udanych testów, na których dany program działa poprawnie. Takie oprogramowanie jest wbudowane w komputery pokładowe samochodów i samolotów, abyśmy wierzyli w niskie prawdopodobieństwo wypadku, natomiast poprawność współbieżnych programów sieci Sleptsova udowadniamy w sposób formalny.

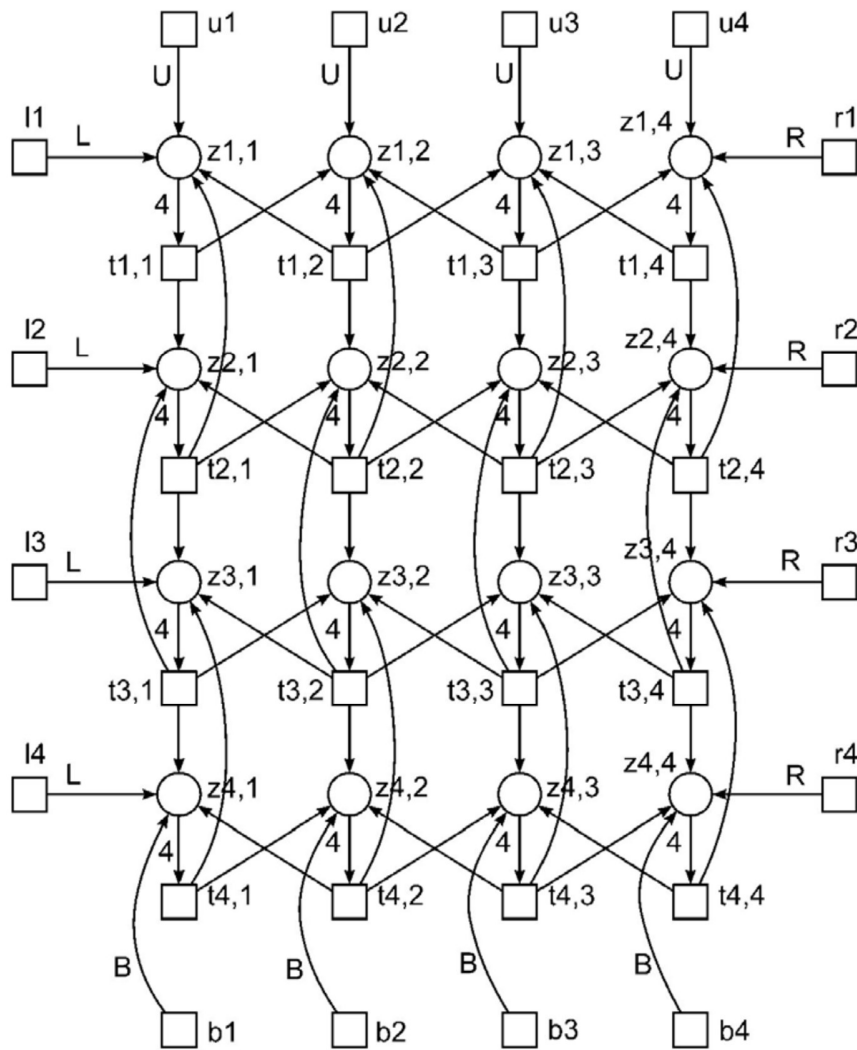
Poniżej zebraliśmy algorytmy sieci Sleptsova dla różnych dziedzin zastosowań:



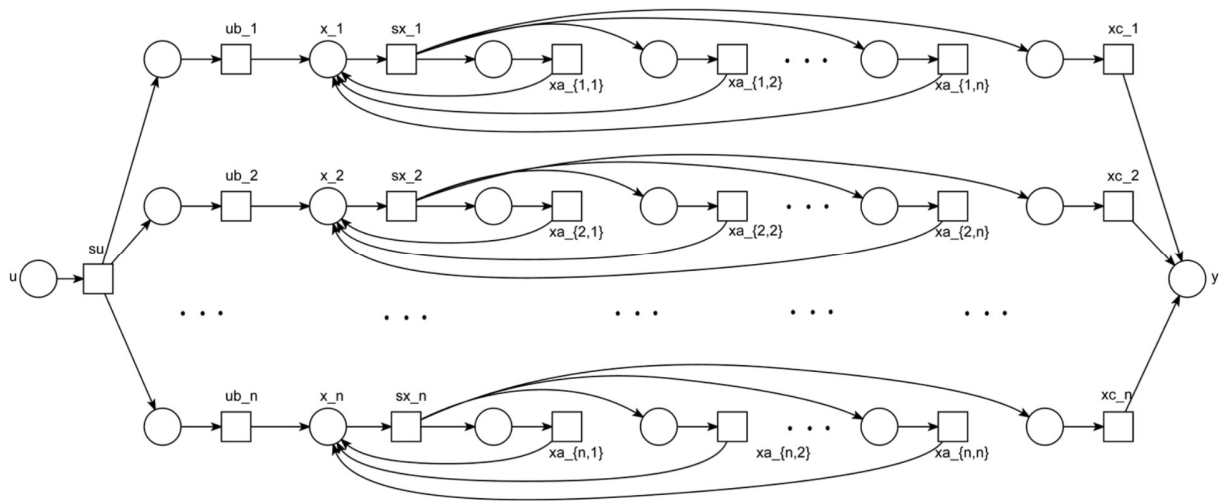
a) Cyberbezpieczeństwo: Szyfrowanie/desyfrowanie RSA z kluczem otwartym;



b) Sterowanie rozmyte: Obliczanie funkcji logiki rozmytej;



c) Numeryczne rozwiązywanie ODE i PDE: Rozwiązywanie równania Laplace'a



d) Sterowanie systemami: Sterowanie liniowe w czasie dyskretnym w dwóch tikach

Przykłady algorytmów sieci Sleptsova dla różnych dziedzin zastosowań

-

Ostatnio moi magiŝtranci zaimplementowali prototypy programowe procesora Sleptsov net oraz kompilatora-linkera opartych na sieciach Sleptsova i zaprezentowali je na międzynarodowej konferencji, publikacja IEEE ich wystąpień planowana jest juŝ wkrótce. Zapraszamy studentów i badaczy z całego ŝwiata do udziału w rozwoju sprzętu komputerowego i technologii programowania Sleptsov net.

Zapraszamy równieŝ inwestorów do rozpoczęcia projektu implementacji przemysłowej paradygmatu obliczeniowego Sleptsov net, w tym implementacji sprzętowej procesora i komputera Sleptsov net. W celu wstępnej integracji z konwencjonalną infrastrukturą, dołączymy go, jako urządzenie rozszerzające, do tradycyjnego komputera, jak to zostało wspomniane w wykładzie dr Jacka Dongarry jako jeden z perspektywicznych kierunków dalszego rozwoju nowoczesnej architektury HPC.

**Sieci Petriego, Salwickiego i Sleptsova jako kontynuacja prac Gilbretha i Gilla**

Ponadto w chwili obecnej, zaplecze projektu SNC zostało dobrze przygotowane poprzez liczne wykłady plenarne i wykłady dedykowane oraz kilkanaŝcie publikacji, w tym w specjalnym wydaniu [Petri/Sleptsov net based technology of programming for parallel, emergent and distributed systems](#) w International Journal of Parallel, Emergent and Distributed Systems.

Niedawno opublikowana praca online [2] pokazuje, ŝe możemy używać wyłącznie sieci Sleptsova (bez angaŝowania innych pojęć) do specyfikacji dowolnych obliczeń. Stanowi to solidne teoretyczne podstawy dla jednorodności SNC.

W artykule przedstawiono równieŝ bezstronny historyczny przegląd rozwoju równoległych schematów programowych. Po pierwsze, schemat procesów równoległych pojawił się we wczesnych pracach Franka

i Lilian Gilbreth z 1921 roku i został znormalizowany w 1947 roku. W 1958 roku Gill zaczął używać dwudzielnych grafów skierowanych do specyfikacji obliczeń równoległych. W 1962 r. Petri dalej rozwija model sieci miejsc i tranzycji dodając znaczniki i regułę odpalania tranzycji. Agerwala i Hack rozszerzają ten model o łuki inhibitorów w latach 70-tych. Salwicki i Sleptsov w latach 80-tych wygenerowali idee odpalania maksymalnej ilości tranzycji równoległych oraz wielokrotnego odpalania tranzycji, rozwinięte i opublikowane w pracach Burkharda i Zaitseva. Turingowskie kompletne sieci przejść reprezentują doskonały graficzny język dla obliczeń współbieżnych, choć działają wykładniczo wolniej niż maszyna Turinga. Wreszcie, sieć Sleptsova naprawia tę wadę, działając szybko i otwierając perspektywy na sprzętową implementację homogenicznego, masywnego, równoległego superkomputera.

Perspektywiczny kierunek jest realizowany w prototypach oczekujących inwestycji na jego pełnowymiarowe wdrożenie. Wykorzystajmy co najmniej 10% ze zmarnowanych 99,2% inwestycji w nowoczesne superkomputery USA (liczba wzięta z Jack Dongarra Turing Award Talk) do projektu wdrożenia SNC, aby uzyskać nowy rekord realnej wydajności obliczeń.

*P.S. W ciągu mojego życia miałem szczęście trafić na najlepszych nauczycieli, a dokładniej miałem szczęście, że oni – czyli Anatoly Sleptsov w 1988 roku i Jack Dongarra w 2017 roku - mnie wybrali na ucznia. Od Anatolija Sleptsova nauczyłem się wiele na temat procesów współbieżnych oraz sieci miejsc i tranzycji, następnie on skierował mnie do praktycznej implementacji naszych teoretycznych wyników w zakładach Topaz, a Opera-Topaz narodziła się w 1990 roku, zanim takie siatki nazwano "workflow", a zasadę przyspieszenia nazwano "exhaustive use of rule". Jack Dongarra wprowadził mnie w cudowny świat superkomputerów i przeprowadził przez proces implementacji mojej teorii klanów na nowoczesnej architekturze równoległej i rozproszonej, wspólnie wydanego oprogramowania, wspólnie opublikowanych prac. Poświęciłem moją [odę](#) Jackowi Dongarrze i chcę wierzyć, że [nasza wspólna praca](#) przestawiła kompas jego badań nieco bardziej w kierunku obliczeń rzadkich.*

### **Źródła**

1. Zaitsev D.A. Sleptsov Nets Run Fast, [IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016, Vol. 46, No. 5, 682 - 693](#)
2. Dmitry A. Zaitsev, Strong Sleptsov nets are Turing complete, [Information Sciences, Volume 621, 2023, Pages 172-182.](#)