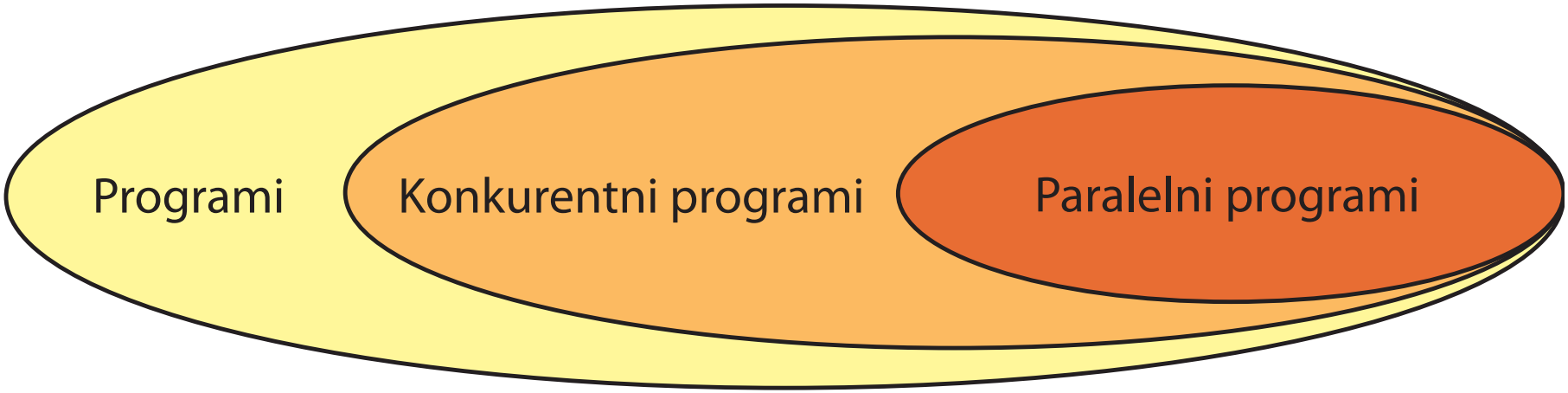
Uvod u paralelno računarstvo

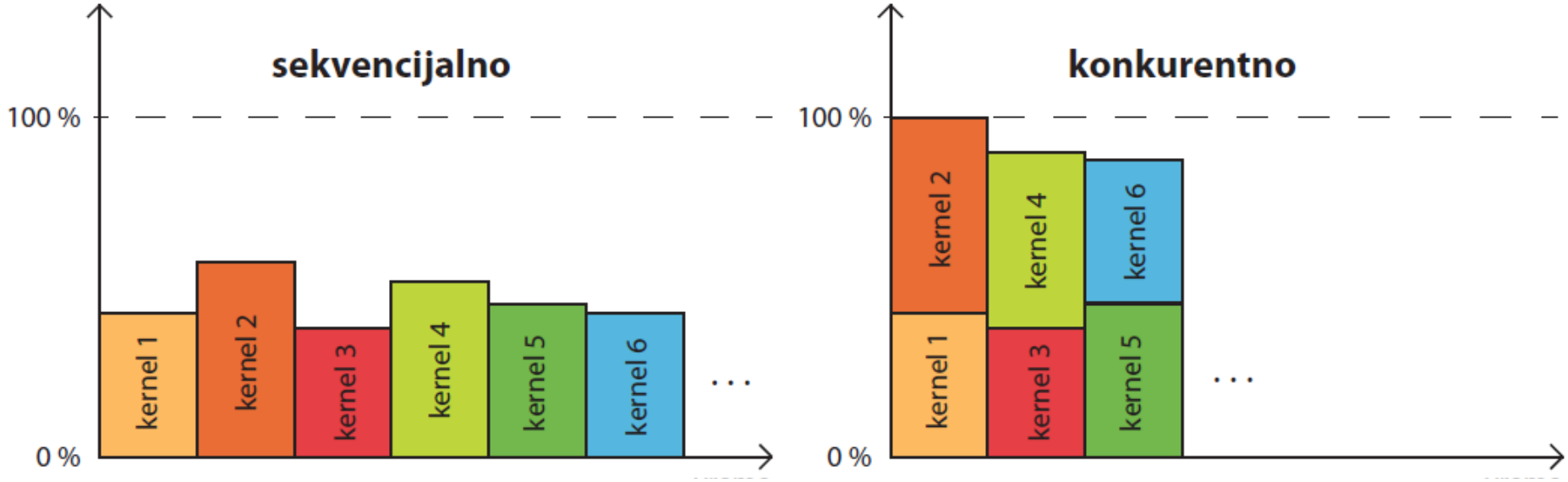
Konkurentna, paralelna i distribuirana obrada

- Odnos programa, konkurentnih programa i paralelnih programa:



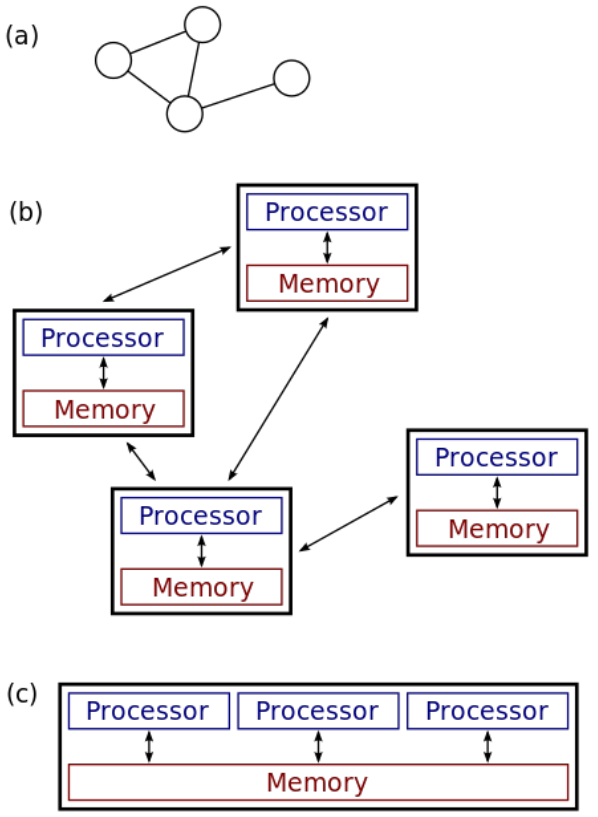
Sekvencijalni i konkurentni programi

- Izvršavanje sekvencijalnih i konkurentnih programa:



Konkurentna, paralelna i distribuirana obrada

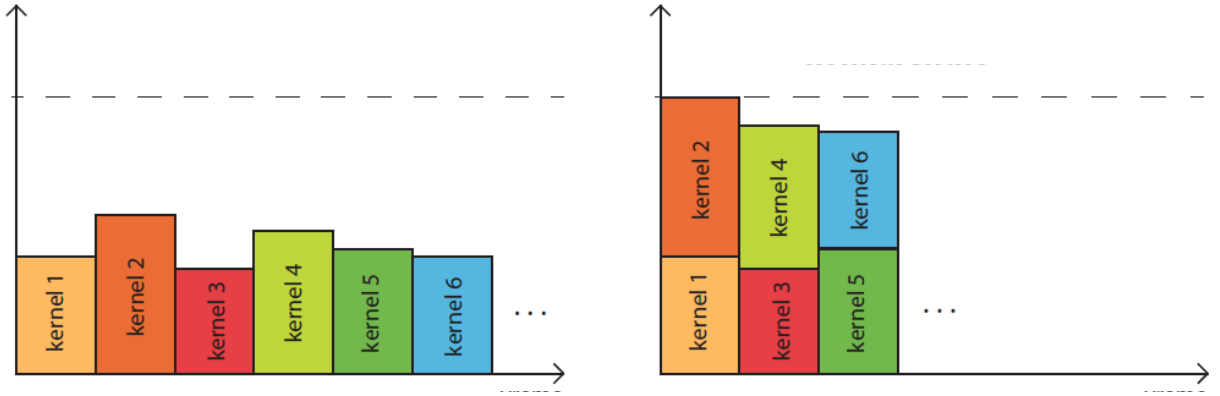
- **Konkurentno** – dešava se **tokom istog vremenskog intervala**
- **Paralelno** – dešava se **u isto vreme**. **Paralelna obrada** po definiciji zahteva **veći broj procesora ili jezgara**. Više od jednog konkurentnog procesa može se **istovremeno** izvršavati. Kao i kod konkurentnog rada, ne mora biti komunikacije niti koordinacije između procesa
- **Distribuirano** – **više programa** izvršava se **konkurentno** i **međusobno komunicira** kako bi zajedno izvršili neko izračunavanje. Suština distribuirane obrade je u tome da se **rezultujuće izračunavanje distribuira između više procesa** koji **međusobno komuniciraju**



Izvor: https://en.wikipedia.org/wiki/Distributed_computing

Tipovi programa i sistema

- **Obrada podataka**, tj. **računarski programi** koji implementiraju odgovarajuće algoritme, po svojoj prirodi mogu biti **sekvencijalni** ili **konkurentni**
- **Okruženje**, tj. **računarski sistem** na kome se neki program izvršava, zavisno od svoje arhitekture i dostupnih resursa može nuditi mogućnost **serijskog** ili **paralelnog izvršavanja** računarskih programa
- Da bi **računarski programi** mogli da iskoriste **mogućnosti paralelnih sistema** neophodno je da imaju **konkurentnu prirodu**



Paralelni sistemi

Hijerarhija apstrakcija

Nivo 5 — Nivo programskih jezika

Prevođenje (kompajler)

Nivo 4 — Nivo asemblerskog jezika

Prevođenje (asembler)

Nivo 3 — Nivo mašine operativnog sistema

Delimična interpretacija (operativni sistem)

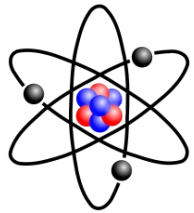
Nivo 2 — Nivo arhitekture skupa instrukcija (ISA)

Interpretacija (mikroprogram)

Nivo 1 — Nivo mikroarhitekture

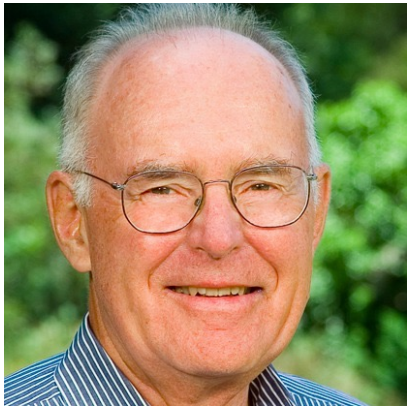
Hardver

Nivo 0 — Nivo digitalne logike

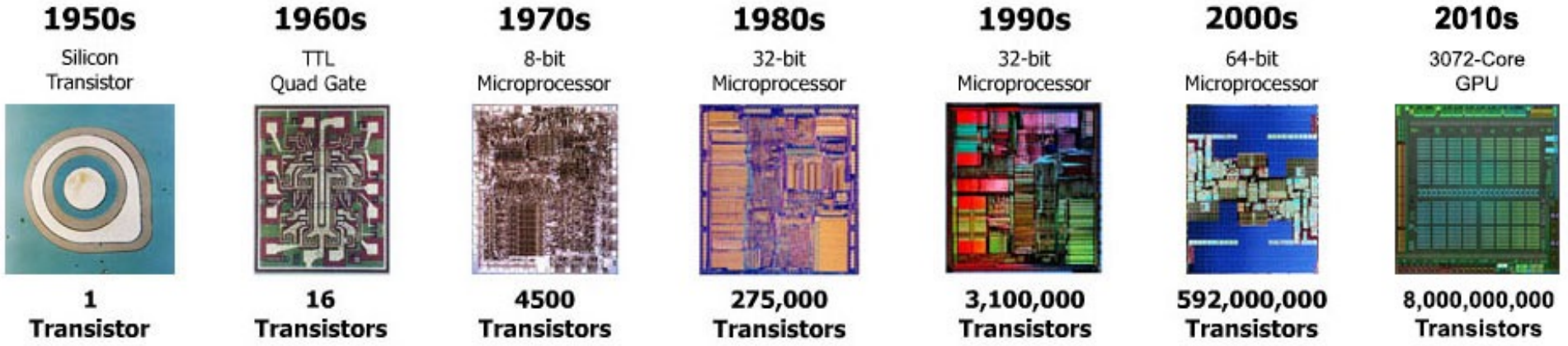


Porast performansi računara

- **Murov zakon** je zapažanje da se broj tranzistora u integrisanim kolima duplira približno svake 2 godine
- **Prestao je nedavno da važi!**
- **Evolucija arhitekture elektronskih računara (1946-danas)**

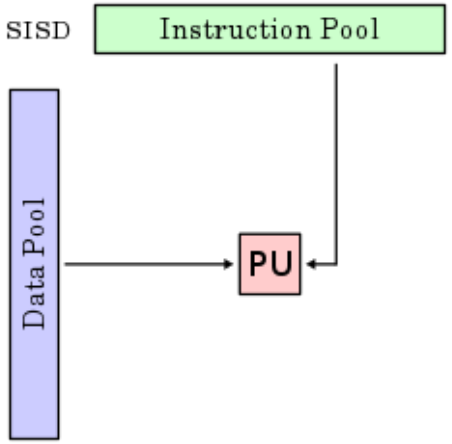


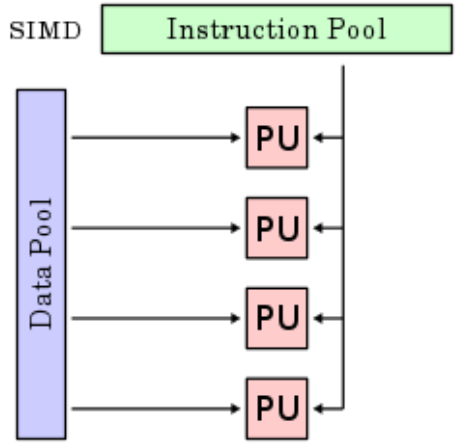
Gordon Moore (1929–2023)



Flynnova taksonomija

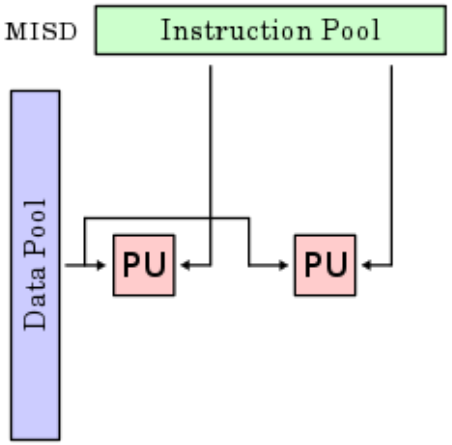
Single Instruction, Single Data (SISD)

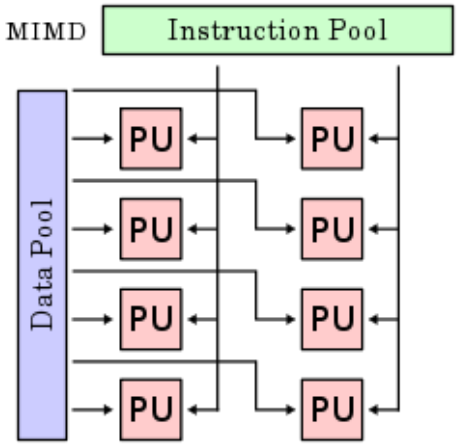




Single Instruction, Multiple Data (SIMD)

Multiple Instruction, Single Data (MISD)

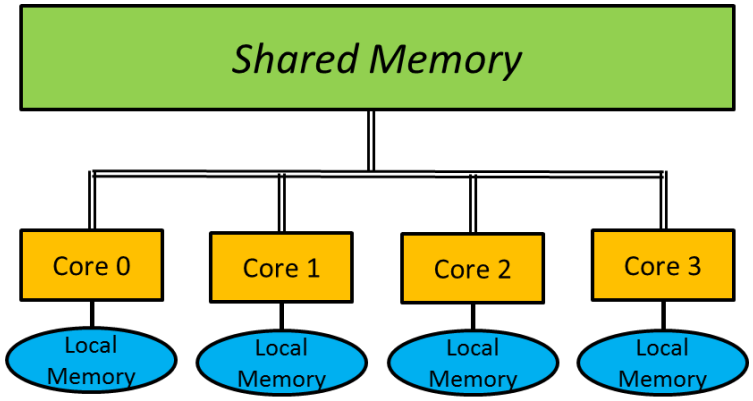




Multiple Instruction, Multiple Data (MIMD)

Izvor: https://en.wikipedia.org/wiki/Flynn%27s_taxonomy

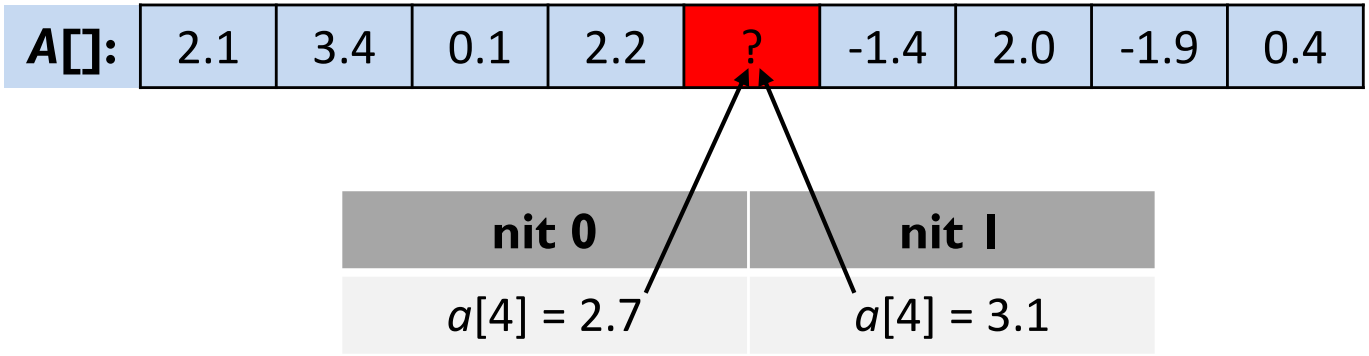
Sistemi sa deljenom memorijom



- **Sva jezgra imaju pristup deljenoj memoriji** preko **zajedničke magistrale** (engl. *shared memory*) – primer višejezgarni CPU
- Pored deljene memorije, svako od jezgara može imati svoju **manju lokalnu memoriju** (npr. keš) kako bi se **smanjio broj skupih memorijskih operacija** (tzv. *von-Neumann bottleneck*)
  - Savremeni višejezgarni CPU sistemi imaju podršku za ostavarivanje **koherentnosti keša** (engl. *cache cocherence*) – engl. *ccNUMA*: *cache coherent non-uniform access architectures*
- Najvažniji programski modeli: **višenitni C++, OpenMP, CUDA**

Izvor: https://parallelprogrammingbook.org/

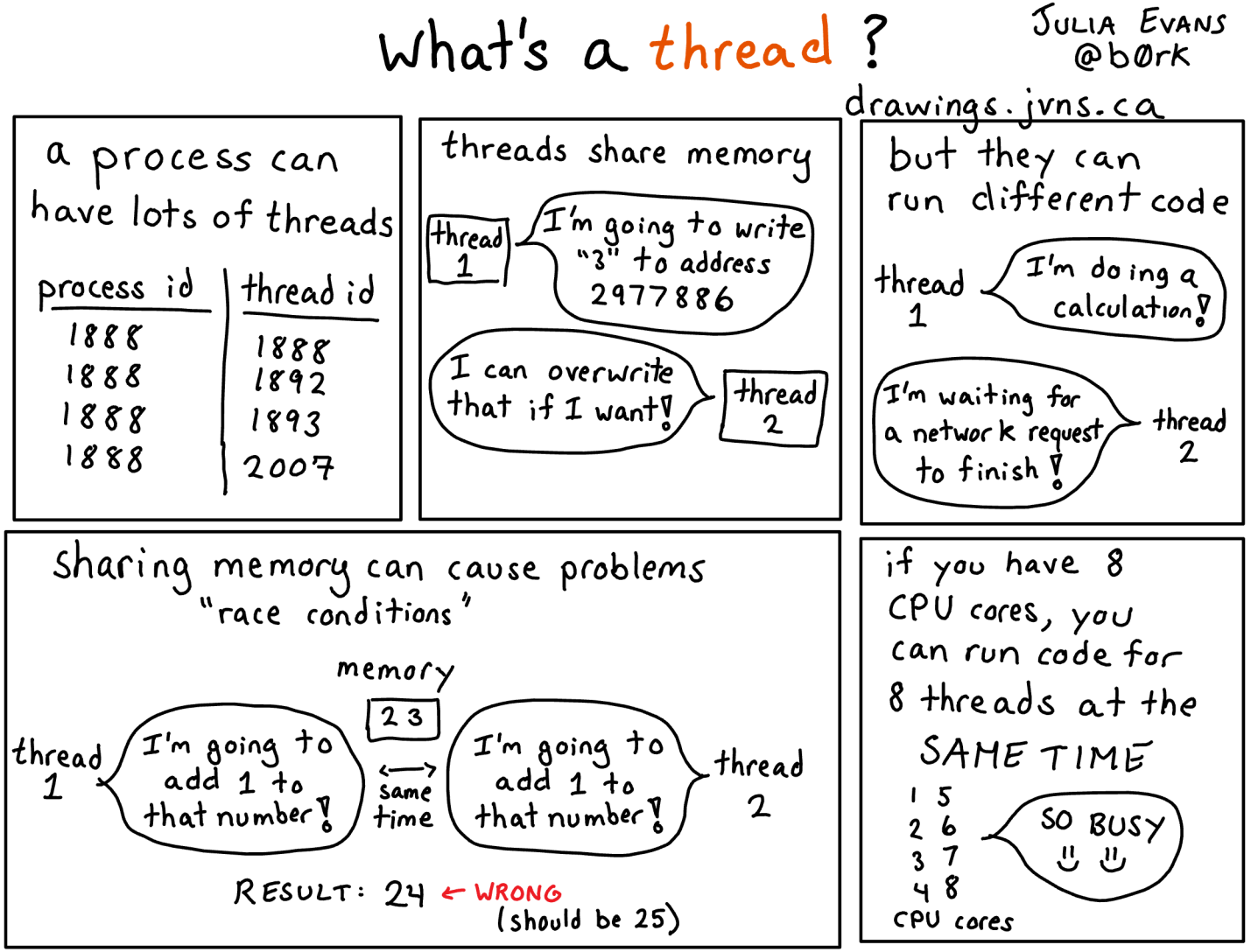
Sistemi sa deljenom memorijom



- Paralelizam se stvara kreiranjem niti koje se konkurentno izvršavaju u sistemu
- Razmena podataka se realizuje tako što niti čitaju sa i pišu u deljene memorijske lokacije
- **Uslov trke** (engl. *race condition*) se javlja kada dve niti istovremeno pristupe deljenoj promenljivoj
  - Odgovarajuće programske tehnike: muteksi, uslovne promenljive, atomične operacije
- Kreiranje niti lakše i brže od procesa – primer iz knjige *Parallel Programming*:
  - CreateProcess(): 12.76 ms
  - CreateThread(): 0.038 ms

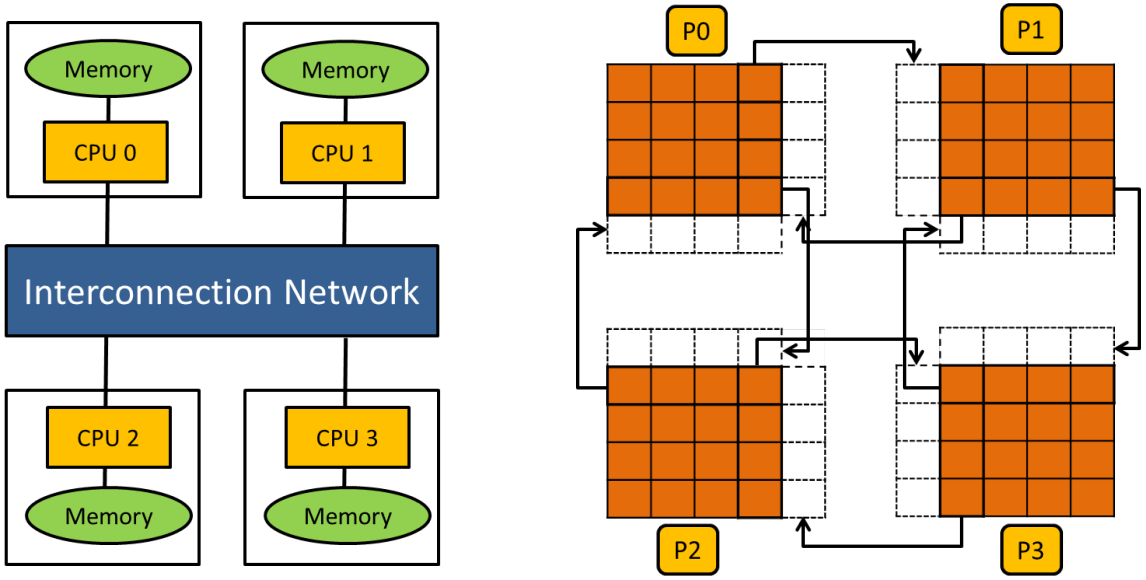
Izvor: https://parallelprogrammingbook.org/

Procesi i niti



Izvor: https://drawings.jvns.ca/drawings/threads.svg

Sistemi sa slanjem poruka

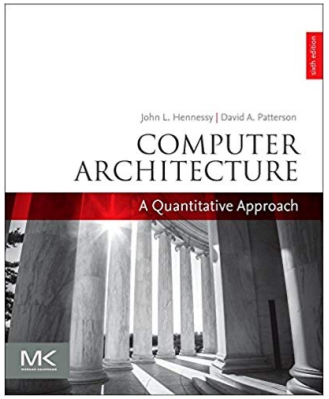


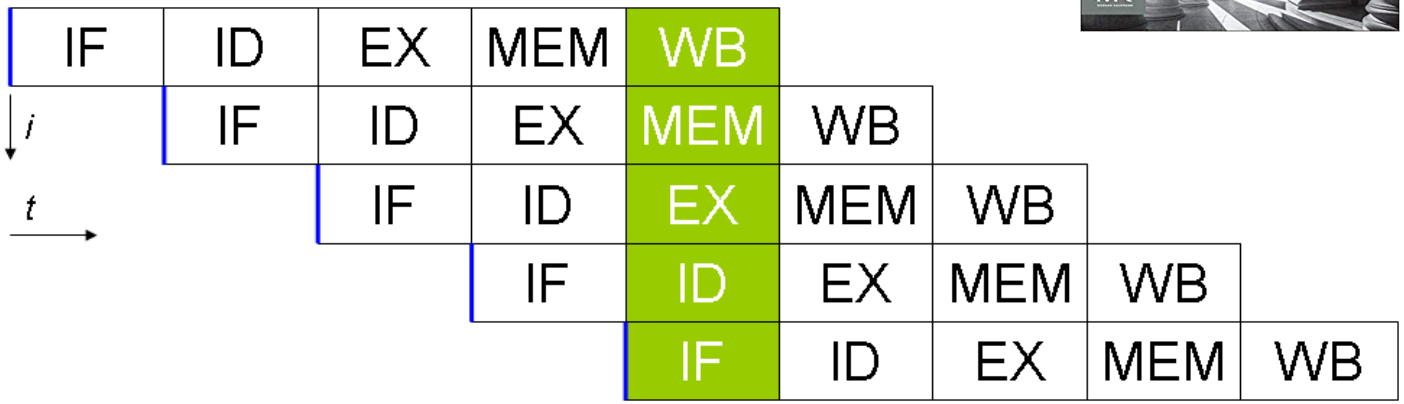
- **Svaki čvor** ima **sopstvenu privatnu memoriju**. Procesori eksplicitno komuniciraju slanjem poruka putem mreže
  - Najpopularniji standard: **MPI** (npr. MPI_Send, MPI_Recv, MPI_Bcast, MPI_Reduce)
- Primer su **računarski** (Beowulf) **klasteri**
  - Skup klasičnih računara povezanih sprežnom mrežom (engl. *interconnection network*) (npr. Ethernet ili Infiniband)

Izvor: https://parallelprogrammingbook.org/

Tipovi paralelizma

- **Paralelizam na nivou bitova** (engl. *bit-level parallelism*)
- **Paralelizam na nivou instrukcija** (engl. *instruction level parallelism* – ILP)
  - Protočna obrada (engl. *pipelining*)
  - Problem zavisnosti po podacima
  - RISC arhitektura
    - prvi primer MIPS – Hennessy, Patterson

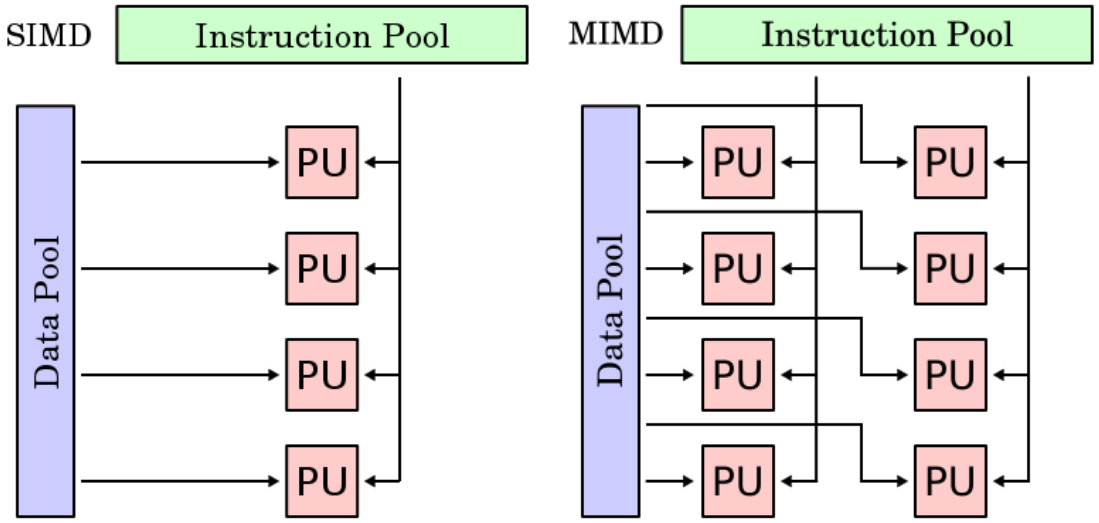




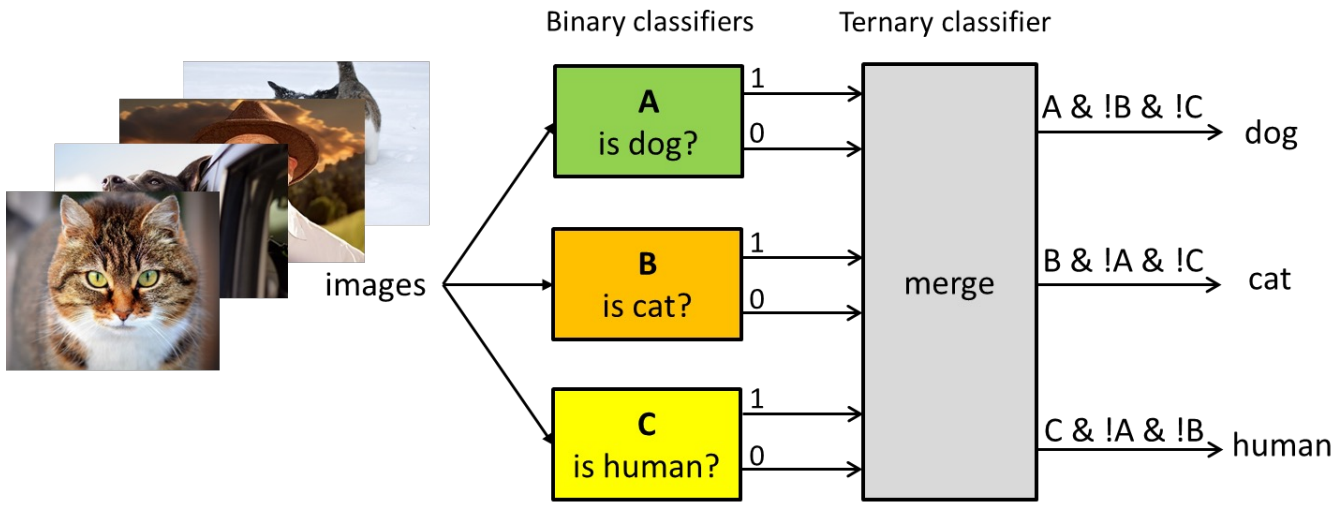
Izvor: https://en.wikipedia.org/wiki/Parallel_computing

Paralelizam na nivou podataka i zadataka

- **Paralelizam na nivou podataka** (engl. *data parallelism*) – tipično za SIMD arhitekture
- **Paralelizam na nivou zadataka** (engl. *task parallelism*)
- **Upleteni paralelizam** (engl. *braided parallelism*) – kombinovani paralelizam na nivoima podataka i zadataka



Paralelizam na nivou podataka i zadataka



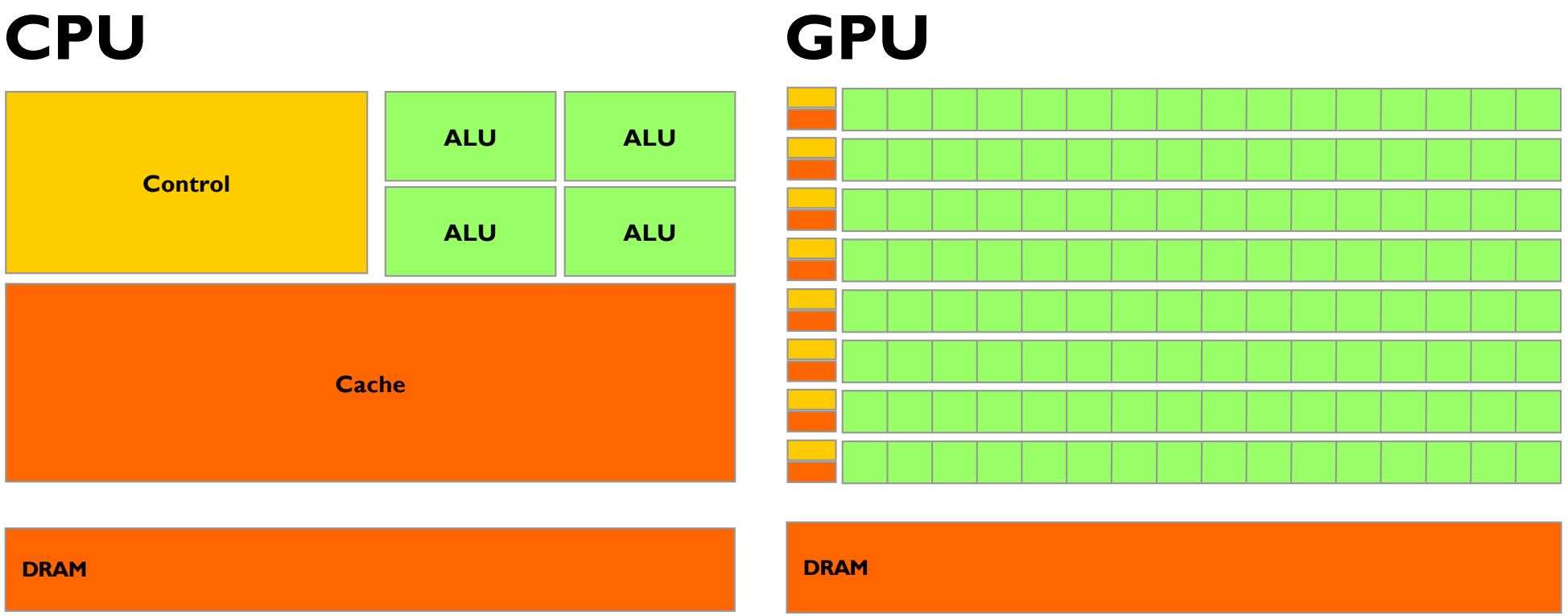
- **Paralelizam na nivou zadataka**
  - Svaki binarni klasifikator pridružen je posebnom procesu (P1, P2, P3)
  - Svaki od procesa klasifikuje svaku od slika primenom pridruženog klasifikatora
  - Rezultati binarne klasifikacije za svaku od slika se šalju P0 koji vrši spajanje
  - Ograničen paralelizam i moguć loš balans opterećenja
- **Paralelizam na nivou podataka**
  - Ulazne slike se dele u grupe
  - Čim proces završi klasifikaciju za dobijenu grupu, planer mu dinamički dodeljuje novu

Izvor: https://parallelprogrammingbook.org/

Heterogeni paralelni računarski sistemi

- Prelazak na **heterogene računarske procesore** je velika promena u **arhitekturama računara** i **računarstvu uopšte**
- **Homogeni računarski sistemi** – jedan ili više procesora iste arhitekture koriste se za izvršavanje programa
- **Heterogeni računarski sistemi** – skup procesora zasnovanih na različitih arhitekturama (CPU, GPU, FPGA, DSP) koristi se za izvršavanje programa
- **Svaki procesor** namenjen je za **različite zadatke** i s toga je njegova arhitektura zasnovana na **različitoj projektnoj filozofiji**
- Izvršavanje zadataka na arhitekturama koje su im najbolje prilagođene vodi do **unapređenih performansi** u smislu vremena i energije**,** ali zahteva **nove tehnike u programiranju** (primer – GPGPU programiranje)

Paralelna obrada na CPU i GPU

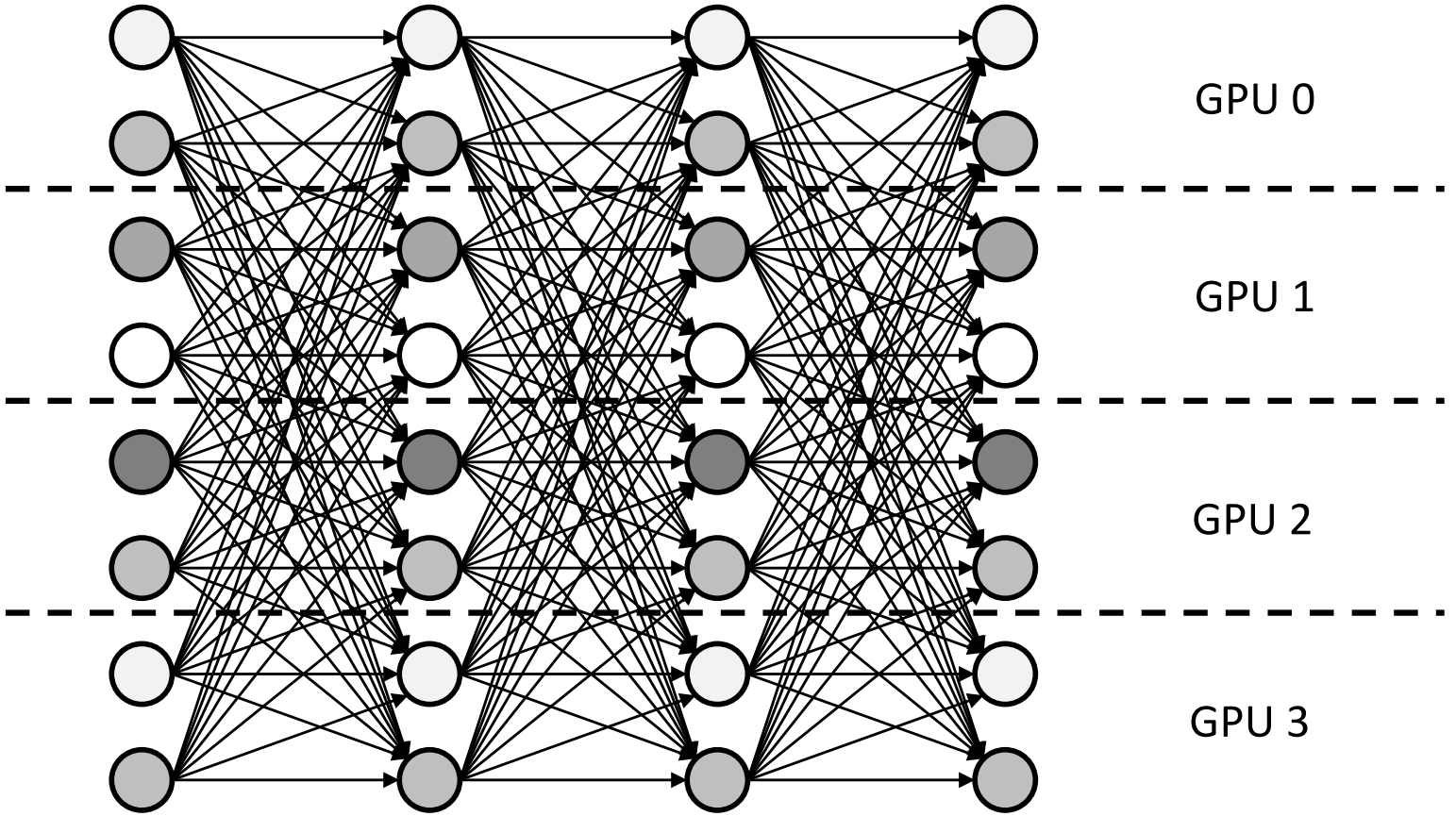


von Neumann (SISD), višejezgarna

SIMD, mnogojezgarna

Izvor: https://commons.wikimedia.org/wiki/File:Cpu-gpu.svg

Paralelizam modela za duboko učenje



Izvor: https://parallelprogrammingbook.org/