Tehnika vertikalnog skaliranja za poboljšanje računalnih performansi

Buljat, Duje

Sveučilište u Zadru

Odjel za informacijske znanosti

Stručni prijediplomski studij

Informacijske tehnologije

Duje Buljat

Tehnika vertikalnog skaliranja za poboljšanje računalnih performansi

Završni rad

Zadar, 2024.

Sveučilište u Zadru
Odjel za informacijske znanosti
Stručni prijediplomski studij
Informacijske tehnologije

Tehnika vertikalnog skaliranja za poboljšanje računalnih performansi

Završni rad

Student/ica:
Duje Buljat

Mentor/ica:
dr. sc. Frane Urem prof. struč. stud.

Zadar, 2024.

Izjava o akademskoj čestitosti

Ja, **Duje Buljat,** ovime izjavljujem da je moj **završni** rad pod naslovom **Tehnika vertikalnog skaliranja za poboljšanje računalnih performansi** rezultat mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na izvore i radove navedene u bilješkama i popisu literature. Ni jedan dio mojega rada nije napisan na nedopušten način, odnosno nije prepisan iz necitiranih radova i ne krši bilo čija autorska prava.

Izjavljujem da ni jedan dio ovoga rada nije iskorišten u kojem drugom radu pri bilo kojoj drugoj visokoškolskoj, znanstvenoj, obrazovnoj ili inoj ustanovi.

Sadržaj mojega rada u potpunosti odgovara sadržaju obranjenoga i nakon obrane uređenoga rada.

Zadar, 10. listopada 2024.

Sažetak

Ovaj rad se bavi analizom performansi višejezgrenih procesora, s posebnim naglaskom na izazove vertikalnog skaliranja, poput memorijskog uskog grla i koherencije predmemorije (cache). Provedeni testovi performansi (benchmark) u virtualnom okruženju pokazuju kako povećanje broja procesorskih jezgri može dovesti do zasićenja performansi zbog konkurentnog pristupa istom memorijskom prostoru. U praktičnom dijelu rada razvijena je web aplikacija koja testira enkripciju i dekripciju s pomoću višejezgrenih procesora, te mjeri opterećenje procesora prije i poslije izvršenja zadataka, čime se prikazuju izazovi skalabilnosti i učinkovitosti.

Osim toga, rad istražuje specijalizirane procesore poput grafičkih procesora (GPU-ova), procesora za neuronske mreže (NPU-ova) i Tensor procesora (TPU-ova). Ovi procesori koriste specifične arhitekture koje nadvladavaju ograničenja tradicionalnih CPU sustava, omogućujući bržu i učinkovitiju obradu podataka u specifičnim primjenama kao što su umjetna inteligencija, duboko učenje, te masivna paralelizacija. Rezultati istraživanja potvrđuju teoretske postavke o ograničenjima tradicionalnog vertikalnog skaliranja i ukazuju na potrebu za primjenom alternativnih arhitektura radi postizanja optimalne računalne učinkovitosti.

Ključne riječi: višejezgreni procesori, cache koherencija, vertikalno skaliranje, performanse procesora, specijalizirani procesori

Sadržaj

POPIS KORIŠTENIH KRATICA

AI - Umjetna inteligencija (Artificial Intelligence)

ASIC - Aplikativno specifičan integrirani sklop (Application-Specific Integrated Circuit)

CPU - Centralna procesorska jedinica (Central Processing Unit)

DMA - Direktan pristup memoriji (Direct Memory Access)

DSP - Procesor za digitalnu obradu signala (Digital Signal Processor)

DVFS - Dinamično skaliranje napona i frekvencije (Dynamic Voltage and Frequency Scaling)

FPGA - Programabilni logički sklopovi (Field-Programmable Gate Arrays)

GPU - Grafička procesorska jedinica (Graphics Processing Unit)

HBM - Memorija s velikom propusnošću (High Bandwidth Memory)

IoT - Internet stvari (Internet of Things)

L1 - Prva razina cache memorije (Level 1 Cache)

L2 - Druga razina cache memorije (Level 2 Cache)

L3 - Treća razina cache memorije (Level 3 Cache)

MAC - Jedinica za množenje i akumulaciju (Multiply-Accumulate Unit)

NPU - Procesor za neuronske mreže (Neural Processing Unit)

RAM - Radna memorija (Random Access Memory)

SMP - Simetrično multiprocesiranje (Symmetric Multi-Processing)

SRAM - Statička memorija s nasumičnim pristupom (Static Random Access Memory)

TPU - Procesor za tenzorske operacije (Tensor Processing Unit)

1. Uvod

U suvremenom računalstvu, rastući zahtjevi za obradom podataka i složenim izračunima zahtijevaju stalna poboljšanja u arhitekturi procesora. Jedan od ključnih pristupa poboljšanju računalnih performansi je vertikalno skaliranje, odnosno povećanje broja procesorskih jezgri unutar jednog sustava. Ovaj pristup omogućava paralelno izvođenje zadataka, što bi teoretski trebalo rezultirati značajnim povećanjem ukupne računalne snage. Međutim, kako se broj jezgri povećava, pojavljuju se i određeni izazovi, poput problema s pristupom memoriji i sinkronizacijom podataka između jezgri.

Višejezgreni procesori dijele iste memorijske resurse, što dovodi do problema s cache koherencijom i memorijskim uskim grlima. Cache koherencija je mehanizam koji osigurava da sve jezgre procesora imaju konzistentne podatke, no održavanje ove konzistencije može dovesti do zastoja i smanjenja ukupne učinkovitosti sustava. Kako broj jezgri raste, tako raste i potreba za koordinacijom između jezgri, što uzrokuje dodatna kašnjenja i smanjenje performansi. Zbog tih ograničenja, linearno povećanje performansi postaje teško ostvarivo, što se može vidjeti kroz mjerenja i analize izvedene u ovom radu.

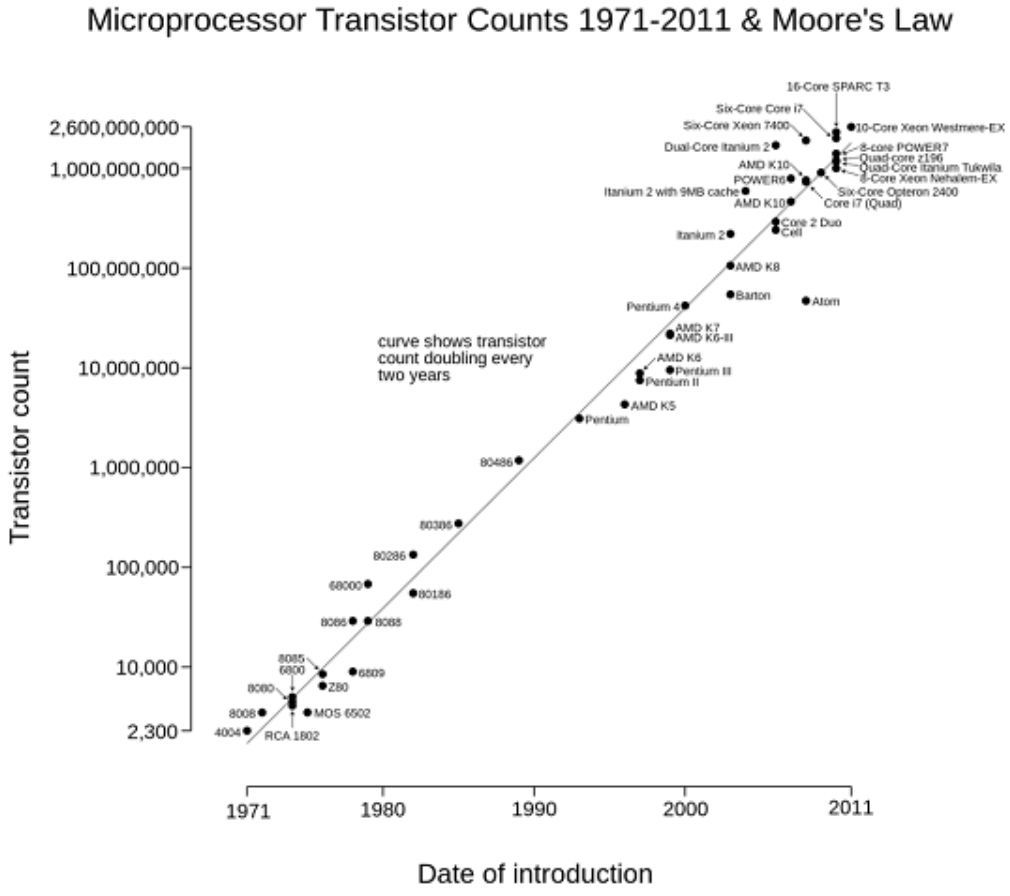
Cilj ovog rada je analizirati performanse višejezgrenih procesora i identificirati ključne izazove povezane s vertikalnim skaliranjem. Kroz testiranja enkripcije i dekripcije unutar virtualnog okruženja s različitim brojem jezgri, prikazana su ograničenja trenutnih arhitektura. Osim toga, rad istražuje specijalizirane procesorske arhitekture poput GPU-ova, NPU-ova i TPU-ova, koje svojim specifičnim dizajnom nadvladavaju ograničenja klasičnih procesorskih sustava.

Praktični dio rada uključuje razvoj web aplikacije koja testira performanse enkripcije i dekripcije koristeći višejezgrene procesore, te mjeri opterećenje svake jezgre prije i poslije izvršenja zadatka. Ova aplikacija omogućava vizualizaciju stvarnog utjecaja vertikalnog skaliranja na performanse sustava, te služi kao temelj za analizu mogućnosti i ograničenja u postojećim procesorskim arhitekturama.

2. Razvoj i izazovi procesorskih arhitektura

2.1. Povijest razvoja procesora i evolucija višejezgrenih procesora

Razvoj centralnih procesorskih jedinica (CPU) započeo je u kasnim 1950-ima s pojavom prvih računalnih sustava temeljenih na vakuumskim cijevima, a kasnije tranzistorima. Mooreov zakon, koji predviđa udvostručavanje broja tranzistora na čipu svake dvije godine, potaknuo je kontinuirani rast performansi procesora kroz povećanje broja tranzistora i taktnih frekvencija sve do ranih 2000-tih godina [1].



Slika 1. Mooreov zakon

Izvor: https://www.aspistrategist.org.au/graph-of-the-week-moores-law/

Kako su arhitekture postajale sve složenije, izazovi vezani uz zagrijavanje i potrošnju energije postajali su sve izraženiji. Ovi problemi su doveli do prelaska na višejezgrene

sustave, gdje se paralelna obrada koristi za povećanje ukupne snage procesora bez povećanja taktne frekvencije. Iako ovaj pristup omogućava veću obradu podataka, pojavili su se izazovi u upravljanju resursima, poput problema cache koherencije, gdje više jezgri pokušavaju pristupiti istim podacima istovremeno, uzrokujući kašnjenja i uska grla u performansama [1].

Uvođenje specijaliziranih procesora poput GPU-ova, NPU-ova i TPU-ova, nastalo je kao odgovor na ove izazove, koristeći paralelizam i optimizaciju za specifične zadatke kako bi se postigla visoka učinkovitost i izbjegla ograničenja koja se susreću kod klasičnih procesorskih arhitektura. Grafički procesori koriste svoje visoko paralelne jezgre za grafičku i znanstvenu obradu, dok NPU-ovi i TPU-ovi služe za optimizaciju operacija strojnog učenja, što omogućava znatno bolje performanse u specifičnim primjenama.

2.2. Tehnološki izazovi i ograničenja skaliranja

Granice skaliranja takta: U ranim fazama razvoja CPU arhitektura, glavni način povećanja performansi bio je povećanje radnog takta procesora. Međutim, povećanje frekvencije takta nije održivo dugoročno zbog nekoliko ključnih problema: zagrijavanja, potrošnje energije i fizičkih ograničenja poluvodičkih materijala. Kako frekvencija raste, tako se povećava i disipacija topline, što zahtijeva složenije i skuplje sustave hlađenja kako bi se izbjeglo pregrijavanje komponenti [2].

Osim toga, povećanje takta uzrokuje višu potrošnju energije, koja raste kvadratno s frekvencijom, što čini takvo rješenje energetski neefikasnim. Ova ograničenja su postala značajna početkom 2000-tih, kada su procesori dostigli točke gdje daljnje povećanje frekvencije nije bilo praktično zbog tih termalnih i energetskih izazova. To je dovelo do prelaska s fokusiranja na povećanje radnog takta na višejezgrene procesore kao glavno rješenje za povećanje računalne snage [2][3].

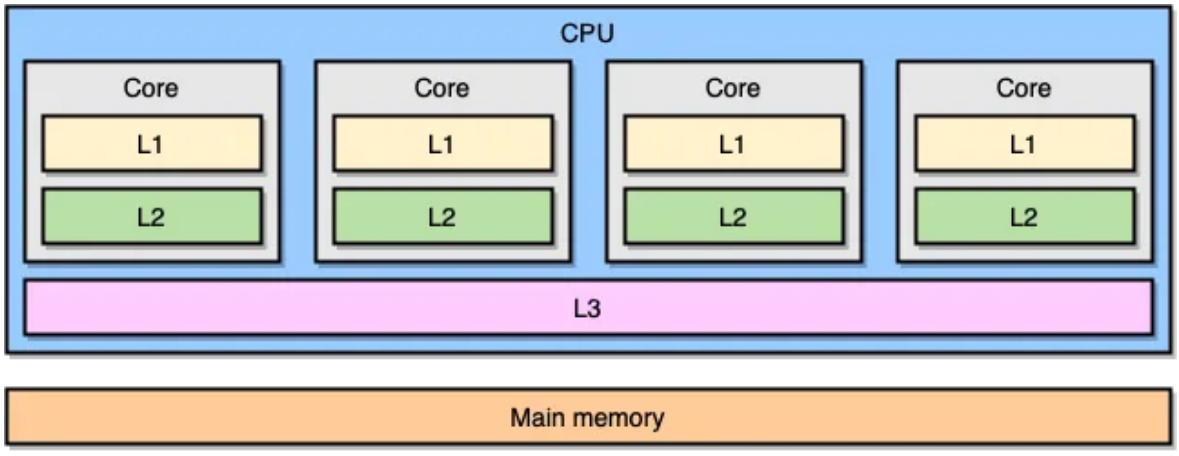
Problemi s paralelizmom: Iako višejezgreni procesori omogućavaju istovremeno izvođenje više zadataka, što teoretski povećava ukupne performanse sustava, pojavljuju se izazovi povezani s paralelnom obradom. Jedan od glavnih problema je složenost programiranja za paralelizam; optimizacija koda za višejezgrene sustave zahtijeva napredne tehnike kako bi se učinkovito podijelili zadaci među jezgrama bez nepotrebnih sukoba ili zastoja.

Drugi izazov je potreba za sinkronizacijom i koherencijom podataka između jezgri. Cache koherencija osigurava da svi procesori imaju konzistentne podatke, ali upravljanje ovim procesima može stvoriti dodatne latencije i smanjiti ukupnu učinkovitost. Višejezgreni sustavi često se suočavaju s problemima "locka" i čekanja kada više jezgri pokušava pristupiti istim resursima, što dovodi do gubitaka u performansama i povećane kompleksnosti sustava [2].

Ovi izazovi naglašavaju potrebu za novim pristupima u dizajnu procesorskih arhitektura, što je potaknulo razvoj specijaliziranih procesora kao što su GPU-ovi, NPU-ovi i ostali, koji su dizajnirani za prevladavanje ograničenja tradicionalnih višejezgrenih CPU sustava kroz drugačije arhitekture i pristupe paralelizmu [3].

2.3. SMP arhitektura i problem koherencije

Moderni procesori baziraju se na konceptu simetričnog multiprocesiranja (SMP). U SMP sistemu, procesor je dizajniran tako da su dvije ili više jezgri spojene na dijeljenu memoriju (RAM). Također postoje tri razine cache-a kako bi se ubrzao pristup memoriji. Cache memorija ključni je element u dizajnu modernih procesora jer omogućava smanjenje latencije pristupa podacima i optimizaciju performansi. Organizirana je u više razina: L1, L2 i L3. L1 cache je najbrža, ali najmanja razina cachea, smještena najbliže jezgrama procesora, često podijeljena na instrukcijski i podatkovni cache. L2 cache je veći i nešto sporiji, služeći kao srednji sloj koji podržava L1 cache s dodatnim prostorom za često korištene podatke. L3 cache, koji se dijeli među jezgrama procesora, je najsporiji i najveći sloj, optimiziran za pristup podacima koji nisu dostupni u L1 i L2 cache-evima, čime se smanjuje učestalost pristupa glavnoj memoriji [4].



Slika 2. SMP arhitektura

Izvor: https://teivah.medium.com/go-and-cpu-caches-af5d32cc5592

Rješenja problema cache koherencije dolaze u software i hardware varijantama. Software je problematičan za većinu slučajeva jer ostavlja programeru odgovornost o koherenciji, tj. programer se unutar svoje aplikacije brine o održavanju sinkronizacije među jezgrama procesora. Hardverski pristup s druge strane osigurava koherenciju kroz fizičku komunikaciju među jezgrama brinući se da je svaka jezgra ažurna u svakom momentu. Dijeli se na dvije glavne vrste: **Snooping/Bus-based protokol (Nadzorni protokol)** i **Directory based protokol (Protokol temeljen na direktoriju)**. U oba slučaja se javlja nekoliko stanja u kojima se pojedini cache može nalaziti, za objašnjenje u primjeru se koristi vrijednost neke varijable:

- **Modified:** Vrijednost varijable je promijenjena unutar ovog cache-a, među ostalim cache-vima ne postoje kopije memorijske lokacije trenutne vrijednosti varijable, stoga vrijednosti ovog cache-a nisu koherentne s glavnom memorijom.
- **Owned:** Označava da je vrijednost varijable „dirty“ (vrijednost je promijenjena) i da je moguće da se nalazi unutar više od jednog cache-a. Cache s ovim stanjem sadrži najnoviju, ispravnu vrijednost. Samo jedna jezgra može držati podatke u ovom stanju, dok ostale jezgre drže taj podatak u Shared stanju.
- **Exclusive:** Vrijednost u ovom cache-u je koherentna sa glavnom memorijom. Ne postoje kopije memorijske lokacije među ostalim cache-evima.
- **Shared:** Vrijednost u ovom cache-u je koherentna s glavnom memorijom. Kopije ove vrijednosti mogu postojati među ostalim cache-evima. Također je samo za čitanje (read-only).
- **Invalid:** Ova vrijednost nije validna.

Ova stanja također čine još nekoliko protokola, uz to što se koriste u već spomenuta dva:

MSI, MESI, MOESI [17].

MSI protocol	MESI protocol	MOESI protocol
MSI is basis of three state(Modified(M),Shared (S), and Invalid (I)).	MESI is basis of four states(Modified(M),Exclusive (E),Shared (S), and Invalid (I)).	MOESI is basis of five states(Modified(M),Owned (O),Exclusive(E),Shared (S), and Invalid (I)).
multiple copies of the block at the same time can do and transition from Shared to Modify can be done without reading data from the cache.	Exclusive (E) added to reduce the number of bus messages sent out for invalid to modified transition.	Owned (O) added to avoid the need of copying back to main memory (write update).
Area utilization of MSI protocol is few by number of use register and flip-flob.	Area utilization of MESI protocol is more as compared to MSI protocol.	Area utilization of MOESI protocol is more as compared to MESI and MSI protocol.

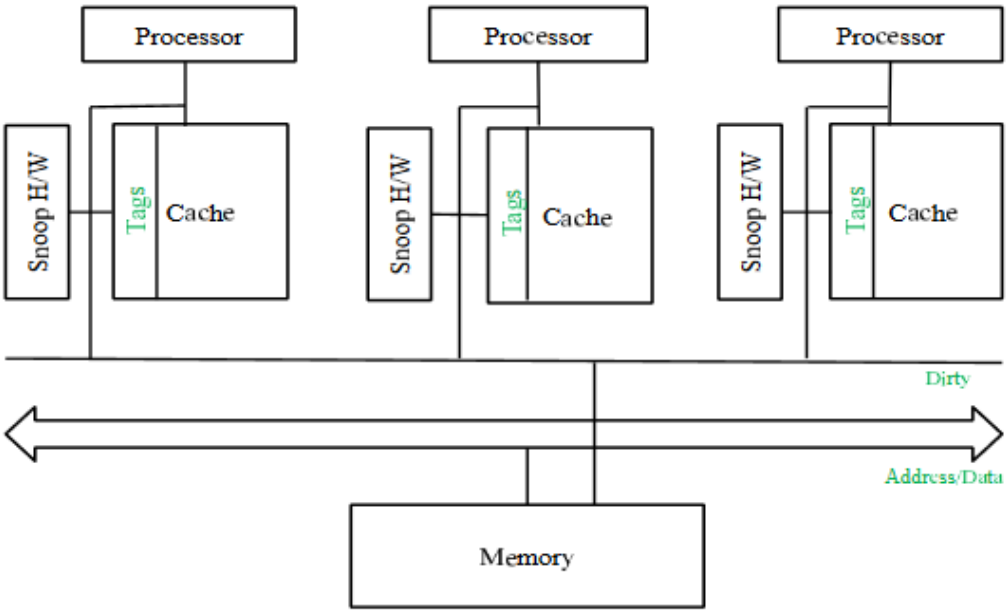
Slika 3. MSI, MESI, MOESI protokoli

Izvor: [17]

Snooping protokol: Svi cache-vi su spojeni kroz jednu, zajedničku sabirnicu (bus), prilikom svake promjene stanja nekog cache-a, obavještavaju se svi ostali cache-evi. Efikasan je kod malog broja međusobno spojenih jezgri, no dolazi do problema kod većeg broja jezgri jer se koriste emitirani (broadcast) signali, koji se šalju svima, stoga kod velikog broja signala poslanih odjednom može doći do zagušivanja sabirnice, gdje se gube performanse. Ovaj protokol nadalje se dijeli na **Write invalidate** i **Write update** kod načina finalne sinkronizacije podataka prije prosljeđivanja u glavnu memoriju.

Primjer: Prilikom pokušaja dohvaćanja vrijednosti varijable A, cache C1 šalje emitirani signal glavnoj memoriji preko zajedničke sabirnice. Taj emitiran signal također dolazi do svih ostalih jezgri, tj. njihovih cache-eva. Ako među ostalim cache-evima ne postoji kopija te varijable, cache koji je zatražio vrijednost varijable dobiva istu i postavlja joj stanje Shared. Cache C2 također dobiva vrijednost varijable A nakon što je zatraži i postavlja stanje Shared. C1 mijenja vrijednost varijable A te mijenja njen status u Modified. U slučaju **Write invalidate** C1 šalje invalidacijski signal ostalim jezgrama kako bi promijenili svoju kopiju vrijednosti varijable A u Invalid. Nakon toga nova vrijednost varijable A prosljeđuje se u

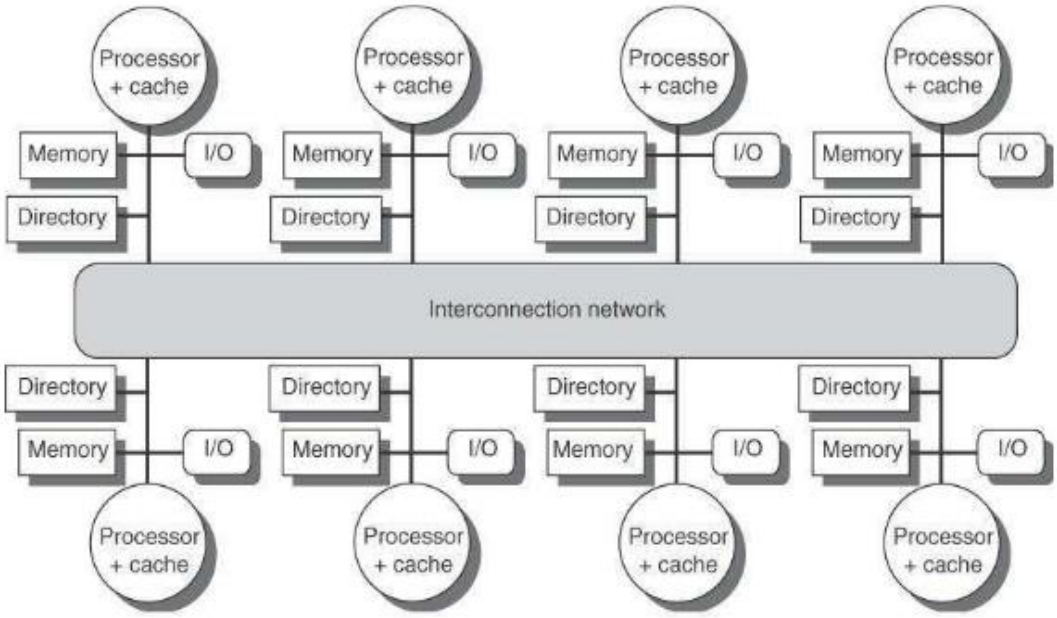
glavnu memoriju. U slučaju **Write update** C1 šalje signal za ažuriranje ostalim jezgrama s novom vrijednosti varijable A kako bi mogli ažurirati svoju kopiju varijable A te je tako ostaviti u Shared stanju, nakon toga nova vrijednost A šalje se dalje u glavnu memoriju.



Slika 4. Snooping protokol

Izvor: [5]

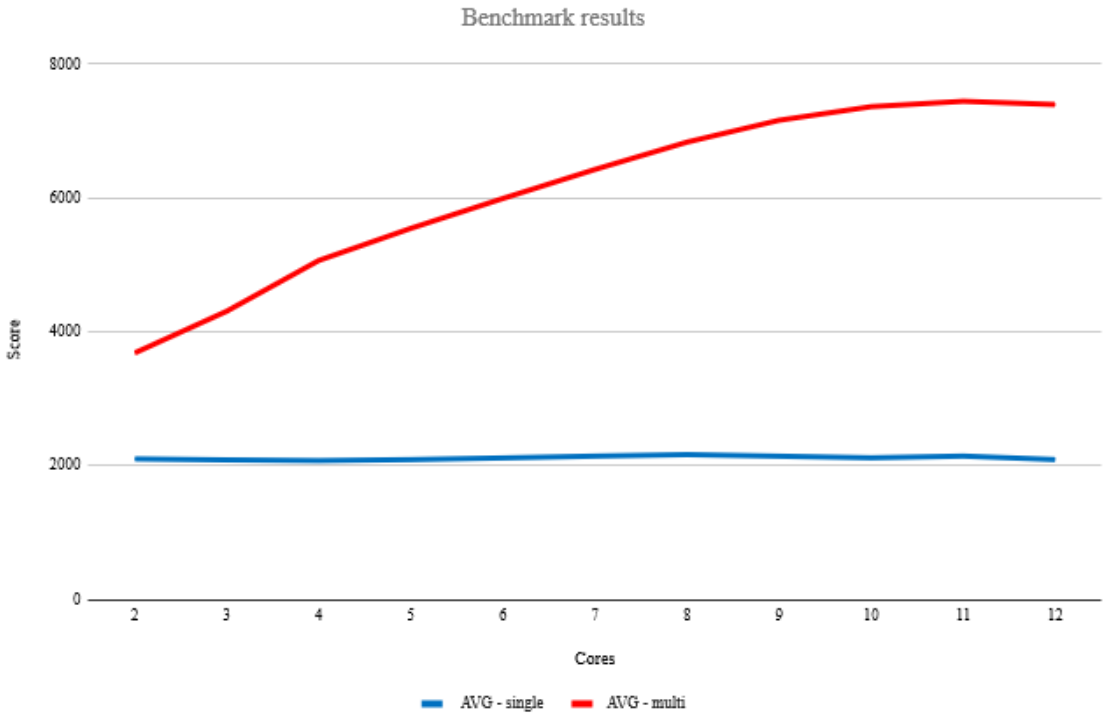
Directory based protokol: Za razliku od prijašnjeg, gdje se informacije izmjenjuju preko jedne zajedničke sabirnice, ovdje se nalazi interkonekcijska mreža (ICN), svaki procesor je svoj čvor (node). Svaki čvor ima svoj direktorij koji omogućuje komunikaciju procesora i glavne memorije preko interkonekcijske mreže. Teže je i kompliciranije za implementirati od prethodnog protokola, no puno je bolji kod skaliranja. Ovdje ne postoje broadcast signali, već pojedini procesor šalje direktan signal nekom drugom procesoru preko ICN-a, što pomaže znatno u slučaju s puno procesora [5].



Slika 5. Directory based protokol

Izvor:[5]

2.4. Analiza performansi višejezgrenih procesora



Slika 6. Performanse procesora u benchmark testiranju

Graf prikazuje rezultate testova performansi procesora u virtualnom okruženju s različitim brojem dostupnih jezgri, od dvije do dvanaest jezgri. Mjerene su performanse procesora u single-core i multi-core modovima kako bi se evaluirala učinkovitost skaliranja performansi s povećanjem broja jezgri. Linija koja predstavlja single-core rezultate ostaje relativno stabilna bez značajnih varijacija, dok multi-core rezultati pokazuju inicijalni rast performansi koji se smanjuje i na kraju stagnira kako se broj jezgri povećava.

Interpretacija Rezultata

Graf prikazuje kako performanse višejezgrenih procesora rastu s povećanjem broja jezgri, no taj rast nije linearan. U rasponu od 2 do 8 jezgri, vidljiv je značajan napredak u multi-core testovima, što sugerira učinkovitu paralelizaciju zadataka. Međutim, između 8 i 12 jezgri, rast performansi se znatno usporava, što je u skladu s ranije opisanim problemima vertikalnog skaliranja.

Uočena stagnacija performansi potvrđuje teorijska razmatranja o memorijskim uskim grlima i problemima cache koherencije. Iako dodatne jezgre povećavaju ukupni broj operacija koje procesor može izvesti, limitirajući faktori, poput sinkronizacije između jezgri i pristupa zajedničkoj memoriji, uzrokuju da povećanje broja jezgri ne rezultira proporcionalnim poboljšanjem performansi.

Zaključak

Rezultati praktičnog testiranja jasno demonstriraju da višejezgreni procesori ne postižu linearno povećanje performansi zbog unutarnjih ograničenja arhitekture. Ovi rezultati naglašavaju važnost optimizacije postojećih arhitektura i istraživanja novih rješenja, kao što su specijalizirani procesori, kako bi se zaobišla ograničenja koja uzrokuju zasićenje performansi pri većem broju jezgri.

3. Specijalizirani procesori kao rješenja za specifične zadatke

3.1. Grafički procesor (GPU)

3.1.1. Arhitektura i funkcionalnost

Grafički procesori (GPU) su dizajnirani za masivnu paralelizaciju, omogućujući istovremenu obradu tisuća niti, što ih čini izuzetno učinkovitim za zadatke poput grafičkog renderiranja, simulacija i znanstvenih izračuna. Moderna arhitektura GPU-a uključuje više razina cache memorije (L1, L2) te globalnu memoriju, optimiziranu za brzo čitanje i pisanje podataka. L1 cache služi za brzu dostupnost podataka unutar jednog streaming multiprocesora (SM), dok L2 cache omogućuje dijeljenje podataka između svih SM-ova na čipu, što smanjuje latenciju pristupa i poboljšava performanse.

Tensor Core jedinice unutar GPU-a dodatno ubrzavaju specifične operacije, poput matriksne multiplikacije, ključno za primjene u strojnome učenju. Evolucija Tensor Core tehnologije, od Volta do Hopper arhitektura, donosi podršku za različite vrste podataka (FP16, BF16, INT8), a novije generacije uvode asinkronu obradu i nove instrukcije (poput wgmma) koje omogućuju izravan pristup iz memorije, čime se smanjuje potreba za upotrebom registara i povećava učinkovitost izvođenja operacija [6].

GPU arhitekture također integriraju podršku za napredne modele upravljanja memorijom, uključujući distribuiranu dijeljenu memoriju, koja omogućuje brži pristup podacima između različitih dijelova procesora, čime se optimizira ukupna propusnost sustava. GPU-ovi koriste tehnike poput L1 i L2 cache pred učitavanja (pre-fetching) kako bi se smanjile latencije prilikom učitavanja podataka iz glavne memorije, što omogućuje neprekinuti tok podataka za paralelne operacije.

3.1.2. Prednosti GPU-a u paralelnoj obradi

Ključna prednost GPU-a je sposobnost paralelne obrade koja omogućuje brže izvršavanje kompleksnih izračuna u usporedbi s tradicionalnim CPU-ovima. Dok CPU-ovi imaju nekoliko snažnih jezgri optimiziranih za serijsku obradu zadataka, GPU-ovi se sastoje od stotina ili tisuća manjih i učinkovitijih jezgri, dizajniranih za istovremenu obradu velikog broja jednostavnih operacija.

GPU-ovi su optimizirani za zadatke poput grafičke obrade, fizikalnih simulacija, strojnih izračuna i procesa dubokog učenja. Njihova arhitektura omogućava učinkovitu obradu masivnih paralelnih operacija, čineći ih superiornim za primjene koje zahtijevaju obradu velike količine podataka u kratkom vremenu. Nadalje, GPU-ovi mogu koristiti asinkrone tehnike kako bi maksimizirali upotrebu resursa bez čekanja, čime se dodatno poboljšava ukupna učinkovitost [6].

Jedan od najvažnijih doprinosa GPU-a u znanstvenom i industrijskom okruženju je njihova sposobnost za ubrzavanje procesa treniranja i inferencije modela umjetne inteligencije (AI). GPU-ovi ubrzavaju izvođenje algoritama strojnog učenja, omogućujući bržu analizu podataka i optimizaciju modela. Tensor Core jedinice unutar GPU-a pružaju dodatne optimizacije za zadatke s niskom preciznošću, kao što su FP16 i INT8 operacije, omogućujući brže izvođenje složenih matriksnih operacija s manjom potrošnjom energije.

3.1.3. Upotreba GPU-ova u znanstvenim izračunima i AI

GPU-ovi su postali ključni alat u znanstvenim istraživanjima, posebno u područjima koja zahtijevaju masivne izračune, poput fizike, kemije, financijskih modela, meteorologije i genetike. Primjerice, u simulacijama molekularne dinamike, GPU-ovi se koriste za modeliranje interakcija između milijuna atoma, omogućujući znanstvenicima da brže i detaljnije proučavaju biološke procese.

U kontekstu umjetne inteligencije, GPU-ovi su neizostavni za treniranje i inferenciju dubokih neuronskih mreža, jer omogućuju bržu obradu velikih skupova podataka. Zbog svoje sposobnosti za paralelizaciju, GPU-ovi mogu simultano obrađivati tisuće matriksnih operacija, što je ključno za zadatke kao što su prepoznavanje slike, obrada prirodnog jezika i predikcija ponašanja korisnika.

GPU-ovi su također našli primjenu u financijskim analizama i predviđanjima, gdje se koriste za brzu obradu velikih količina tržišnih podataka. Njihova mogućnost za paralelnu obradu omogućuje brže simulacije i optimizacije financijskih modela, čime se smanjuje vrijeme donošenja odluka u visokofrekventnom trgovanju i upravljanju rizicima.

Zbog svih navedenih sposobnosti, GPU-ovi nastavljaju igrati ključnu ulogu u razvoju naprednih tehnologija i pružaju temelj za buduća istraživanja i inovacije u znanstvenom, industrijskom i tehnološkom svijetu.

3.2. Procesori za neuronske mreže (NPU)

3.2.1. Specifičnosti NPU arhitekture

Neural Processing Units (NPUs) su specijalizirani procesori dizajnirani za optimizaciju zadataka vezanih uz neuronske mreže i duboko učenje, što ih čini ključnim dijelom moderne infrastrukture umjetne inteligencije. NPU-ovi se razlikuju od općenitih procesora kao što su CPU i GPU, jer su usmjereni na specifične matematičke operacije kao što su matriksna množenja i konvolucije, koje su osnovne operacije u modelima dubokog učenja. Njihova arhitektura uključuje dva glavna računalna elementa: matrični motori (Matrix Engines - MEs), koji izvode sistoličke matriksne multiplikacije, te vektorske motore (Vector Engines - VEs), koji obavljaju generičke vektorske operacije [7].

Ključna komponenta NPU arhitekture je sposobnost za paralelnu obradu velikih skupova podataka, omogućena kroz višeslojnu memorijsku hijerarhiju, poput on-chip SRAM-a za brzi pristup podacima i off-chip HBM-a za masivnu propusnost memorije [7].

Uz to, NPU-ovi koriste sistoličke nizove za optimizaciju matriksne multiplikacije, što omogućuje bržu i učinkovitiju obradu modela dubokog učenja, osobito u treniranju velikih neuronskih mreža.

Novi NPU-ovi, poput Intelovih Neural Compute Engines koji su integrirani u Intelove procesore, koriste skalabilnu arhitekturu s više čipova. Ova arhitektura kombinira specifične akceleratorske jedinice za AI zadatke s vektorskim procesorima za općenitiju paralelizaciju [8].

NPU-ovi također koriste direktan pristup memoriji (DMA) za prijenos podataka između memorijskih regija, smanjujući latencije i poboljšavajući učinkovitost prijenosa podataka [7].

3.2.2. Optimizacija za AI i strojno učenje

NPU-ovi su optimizirani za zadatke koji zahtijevaju visoko paralelne operacije, kao što su treniranje i inferencija neuronskih mreža. Glavna prednost NPU-ova leži u njihovoj mogućnosti da brzo obrađuju velike količine podataka koristeći tehnike kao što je reducirana preciznost. Operacije s 8-bitnim i 16-bitnim brojkama (umjesto standardnih 32-bitnih ili 64-bitnih) smanjuju zahtjeve za memorijom i povećavaju brzinu obrade, dok su rezultati i dalje dovoljno precizni za većinu aplikacija [7].

Uz to, NPU-ovi podržavaju dinamičko skaliranje, omogućujući prilagodbu broja korištenih računalnih jedinica (MEs i VEs) ovisno o zadatku. Ova mogućnost je posebno važna za zadatke poput treniranja neuronskih mreža, gdje je potrebno kontinuirano prilagođavati resurse kako bi se maksimalno iskoristila propusnost sustava. Primjerice, u nekim modelima, poput ResNet-a koji je ME-intenzivan, može doći do nedovoljno iskorištavanja VE-ova, što zahtijeva optimizaciju kroz softverske alate kao što su napredni kompilatori i virtualizacija resursa [7].

Moderne NPU arhitekture koriste se i u rubnom računalstvu (edge computing), gdje pružaju sposobnosti umjetne inteligencije uređajima s ograničenim resursima, poput pametnih telefona ili IoT uređaja. Samsungovi NPU-ovi, na primjer, omogućuju rubnim uređajima da izvode operacije poput prepoznavanja govora ili slike pri vrlo niskoj potrošnji energije [7].

Jedan od ključnih čimbenika njihove efikasnosti je optimizacija potrošnje energije, gdje NPU-ovi postižu visoke performanse uz minimalnu potrošnju zahvaljujući niskonaponskim operacijama i naprednoj arhitekturi za upravljanje memorijom.

3.2.3. Virtualizacija i korištenje u oblaku

Osim svoje upotrebe u rubnim uređajima, NPU-ovi igraju ključnu ulogu u obradnim centrima i oblaku. Njihova sposobnost za virtualizaciju omogućuje dijeljenje hardverskih resursa između više korisnika ili zadataka, što je posebno korisno u sustavima koji podržavaju multi-tenant infrastrukturu. Virtualizacija NPU-ova omogućuje bolju raspodjelu resursa i fleksibilnije korištenje hardverskih jedinica, ovisno o zahtjevima zadataka [7].

U kontekstu oblaka, NPU-ovi omogućuju treniranje složenih modela dubokog učenja, kao što su BERT ili ResNet, gdje mogu upravljati masivnim skupovima podataka i paralelno izvoditi inferencije na više sklopova. Njihova arhitektura također omogućuje smanjenje troškova energije i hardvera, jer su optimizirani za izvršavanje specifičnih AI zadataka s minimalnim korištenjem resursa [7].

3.3. Tensor procesori (TPU)

3.3.1. Razvoj i primjene TPU-a

Tensor Processing Units (TPUs) su specijalizirani procesori razvijeni od strane Googlea s ciljem ubrzavanja zadataka vezanih uz duboko učenje (Deep Learning). TPU je zapravo aplikativno specifičan integrirani sklop (ASIC) koji je dizajniran da optimizira procese vezane uz umjetne neuronske mreže, posebno kod matriksne aritmetike kao što su operacije množenja matrica (matriksno množenje), koje su ključne u algoritmima dubokog učenja. TPU-ovi su prvi put predstavljeni 2015. godine, a implementirani su u Googleove sustave kao što su Google Translate i Google Photos, čime su omogućili brže treniranje i inferenciju neuronskih mreža.

TPU arhitektura koristi sistoličke nizove za izvršavanje masivnih paralelnih operacija. Ova arhitektura omogućava brži pristup podacima i smanjuje latencije kroz slojevitu memorijsku hijerarhiju koja uključuje specifične module za pristup memoriji (Unified Buffer i Weight Memory).[9]

Primjerice, TPU v1 je imao specifičnu MAC jedinicu (Multiply Accumulate Unit) koja je mogla obraditi do 92 TeraOps u sekundi (TOPS), što je omogućilo značajno ubrzanje u odnosu na općenite procesore poput CPU-a ili GPU-a.

TPU-ovi su ključni za optimizaciju modela strojnog učenja u cloud okruženjima. Zahvaljujući integraciji s platformama kao što je TensorFlow, TPU-ovi omogućuju jednostavniji prijelaz korisnika na upotrebu specifičnih hardverskih resursa za treniranje velikih modela neuronskih mreža. Google Cloud TPU-ovi, posebice, omogućuju dinamično skaliranje resursa, gdje se korisnicima omogućuje prilagodba broja TPU čipova ovisno o zahtjevima treniranja modela [9].

3.3.2. Prednosti TPU-a u AI i dubokom učenju

TPU-ovi su razvijeni prvenstveno za ubrzavanje zadataka koji se odnose na treniranje i inferenciju dubokih neuronskih mreža. Oni donose nekoliko ključnih prednosti u odnosu na tradicionalne procesore:

1. Brzina i paralelizacija: Zahvaljujući svojoj sistoličkoj arhitekturi, TPU-ovi mogu simultano obrađivati velik broj matriksnih operacija, što je ključno za treniranje neuronskih mreža. Na primjer, TPU v3 može doseći do 420 TeraOps u sekundi (TOPS), što omogućuje brže treniranje velikih modela dubokog učenja poput BERT-a i ResNet-a [9]. Ova razina performansi omogućuje ubrzanje procesa treniranja i smanjenje vremena potrebnog za obuku modela.
2. Energetska efikasnost: TPU-ovi su optimizirani za visoku učinkovitost u odnosu na potrošnju energije (TOPS/W). To ih čini posebno pogodnima za implementaciju u cloud okruženjima gdje je energetska učinkovitost ključna za optimizaciju troškova rada podatkovnih centara. U usporedbi s GPU-ovima, koji su također snažni akceleratori za AI zadatke, TPU-ovi često pružaju bolje omjere performansi po vatu, što ih čini poželjnima u velikim skalabilnim aplikacijama [9].
3. Jednostavnost korištenja i skalabilnost: Integracija TPU-a s TensorFlow-om omogućuje korisnicima da lakše optimiziraju svoje AI aplikacije bez potrebe za velikim promjenama u kodu. Google Cloud TPU platforma nudi korisnicima mogućnost da skaliraju svoje resurse ovisno o potrebi, što omogućuje lakši prijelaz s lokalnog treniranja na masivno distribuirano treniranje modela dubokog učenja [9].
4. Primjena u rubnom računalstvu (Edge Computing): Manje verzije TPU-a, poput Edge TPU-a, omogućuju korištenje tehnologije umjetne inteligencije u uređajima s ograničenim resursima, kao što su IoT uređaji i pametni telefoni. Ove verzije TPU-ova omogućuju inferenciju neuronskih mreža pri niskoj potrošnji energije, što ih čini pogodnima za aplikacije koje zahtijevaju real-time obradu podataka [9].

Ukratko, TPU-ovi donose značajna poboljšanja u brzini i efikasnosti dubokog učenja, omogućujući korisnicima da brže treniraju složene modele i efikasnije ih primijene u širokom rasponu aplikacija, od cloud računalstva do rubnih uređaja.

3.4. Programabilni logički sklopovi (FPGA)

3.4.1. Fleksibilnost i programabilnost FPGA-a

Field-Programmable Gate Arrays (FPGA) nude veliku razinu fleksibilnosti i mogućnost reprogramiranja, što ih čini jedinstvenima u usporedbi s drugim čipovima kao što su CPU i GPU. Za razliku od standardnih procesora s fiksnim arhitekturama, FPGA-ovi omogućuju korisnicima da dinamički mijenjaju konfiguraciju logičkih sklopova unutar čipa kako bi optimizirali zadatke prema specifičnim potrebama. Ova prilagodljivost čini FPGA-ove izuzetno vrijednima u aplikacijama gdje je potrebna visoka razina optimizacije za određene zadatke. Na primjer, inženjeri koriste alate kao što su Vivado HLS i Intel HLS Compiler kako bi transformirali modele umjetne inteligencije u kod kompatibilan s FPGA arhitekturom, omogućujući napredne prilagodbe za specifične zadatke poput strojne obrade podataka i umjetne inteligencije [10].

FPGA-ovi su posebno korisni za aplikacije koje zahtijevaju nisko kašnjenje i visoku propusnost podataka. Zahvaljujući mogućnosti izravnog prilagođavanja logičkih elemenata, FPGA-ovi se mogu optimizirati kako bi postigli minimalne latencije u komunikaciji s memorijom i procesorima. Na primjer, FPGA-ovi omogućuju paralelno izvršavanje više zadataka u stvarnom vremenu, prilagođavajući hardver specifičnim operacijama bez dodatnih softverskih slojeva, što je značajno u aplikacijama poput obrade slike u stvarnom vremenu [11].

Još jedan ključni aspekt fleksibilnosti FPGA-ova leži u njihovoj energetskoj učinkovitosti. FPGA-ovi koriste tehnike poput dinamičnog skaliranja napona i frekvencije (DVFS) kako bi se prilagodili trenutnim radnim zahtjevima. Ova tehnika omogućuje čipovima da dinamički prilagode napone i frekvencije, smanjujući potrošnju energije kada su operacije manje intenzivne. S obzirom na sve veći naglasak na smanjenje potrošnje energije u modernim aplikacijama, ovo čini FPGA-ove važnim rješenjem, osobito u aplikacijama za rubno računalstvo i mobilne uređaje [10].

3.4.2. Upotreba u specifičnim industrijama

FPGA-ovi su pronašli primjenu u brojnim industrijama koje zahtijevaju specifične hardverske prilagodbe. U telekomunikacijama, oni omogućuju ubrzavanje obrade podataka i prilagodbu mrežnim protokolima u stvarnom vremenu. Zahvaljujući svojoj mogućnosti reprogramiranja, FPGA-ovi se mogu koristiti za enkodiranje i dekodiranje signala, optimizaciju propusnosti mreže i smanjenje kašnjenja u prijenosu podataka. Ova tehnologija je ključna za razvoj 5G mreža, gdje se zahtijeva brza i prilagodljiva obrada podataka [10].

U financijskoj industriji, FPGA-ovi su našli svoju primjenu u ubrzavanju financijskih simulacija i algoritamskog trgovanja. Oni omogućuju brzu obradu velikih količina podataka u stvarnom vremenu, što je ključno u visoko frekventnom trgovanju, gdje milisekunde odlučuju o profitu ili gubitku. Zbog njihove sposobnosti da optimiziraju algoritme na hardverskoj razini, FPGA-ovi pružaju bržu obradu podataka uz minimalno kašnjenje, što ih čini superiornima u aplikacijama koje zahtijevaju brze odluke na temelju analize velikih količina podataka [12].

FPGA-ovi se također koriste u industrijskoj automatizaciji, gdje omogućuju kontrolu strojeva i sustava za nadzor kvalitete. U automatiziranim proizvodnim sustavima, FPGA-ovi se koriste za brzu analizu podataka sa senzora u stvarnom vremenu, omogućujući proizvodnim tvrtkama da optimiziraju procese proizvodnje i osiguraju visoku razinu kvalitete. Prilagodljivost FPGA-a omogućuje im da se koriste za obradu slike, prepoznavanje obrazaca i nadzor proizvodnih linija [11].

3.4.3. Energetska učinkovitost i primjena u rubnom računalstvu

FPGA-ovi su idealni za rubno računalstvo zbog svoje visoke energetske učinkovitosti i fleksibilnosti. Rubno računalstvo zahtijeva lokalnu obradu podataka na uređajima s ograničenim resursima, a FPGA-ovi su prilagođeni za ove scenarije jer omogućuju dinamičko skaliranje performansi uz minimalnu potrošnju energije. Ova energetska učinkovitost postiže se kroz tehnike kao što su clock gating, koja smanjuje potrošnju energije isključivanjem neaktivnih dijelova čipa, te adaptivno upravljanje energijom, gdje se potrošnja energije prilagođava stvarnim zahtjevima zadatka [11].

Jedan od primjera primjene FPGA-ova u rubnom računalstvu je u IoT uređajima, gdje se koristi za obavljanje složenih zadaća poput prepoznavanja uzoraka i analize podataka sa senzora. Ovi uređaji često rade s ograničenim resursima, a FPGA-ovi omogućuju njihovu optimizaciju za specifične zadatke, produžujući vijek trajanja baterija i poboljšavajući ukupnu učinkovitost sustava. Dodatno, FPGA-ovi omogućuju kontinuiranu prilagodbu, jer se mogu reprogramirati za nove zadatke bez potrebe za zamjenom hardvera [10].

3.5. Procesori za digitalnu obradu (DSP)

3.5.1. Primjene u audio i video obradi

Digitalni signalni procesori (DSP) igraju ključnu ulogu u obradi audio i video signala, pružajući visokokvalitetnu obradu u stvarnom vremenu. DSP procesori koriste se u raznim aplikacijama, uključujući aktivno poništavanje buke (ANC), upravljanje glasnoćom poziva, te uklanjanje odjeka. U svim ovim aplikacijama, DSP omogućuje brzu analizu i prilagodbu digitalnih signala, što rezultira poboljšanom kvalitetom zvuka i videa. Na primjer, kod slušalica koje koriste ANC, DSP obrađuje signal koji snima vanjske zvukove, invertira fazu zvuka i poništava pozadinsku buku prije nego što se signal prevede u analognu formu putem digitalno-analognog pretvarača (DAC) [13][14].

U video obradi, DSP procesori se koriste za poboljšanje kvalitete videa kroz procese kao što su filtriranje buke, obnova degradiranih snimaka i skaliranje rezolucije. Tehnologija DSP omogućuje učinkovito obnavljanje starih ili oštećenih videozapisa, čime se smanjuju artefakti i poboljšava opća kvaliteta slike. DSP procesori također omogućuju napredne metode kompresije videa, što značajno smanjuje prostor za pohranu, ali bez gubitka kvalitete, što ih čini ključnima u aplikacijama kao što su streaming usluge i video konferencije [13].

Audio produkcija također značajno ovisi o DSP tehnologijama. Digitalni audio procesori (uobičajeni u aplikacijama za glazbenu produkciju) koriste DSP algoritme za ekvilizaciju, kompresiju, dodavanje efekata kao što su reverb i delay, te za miješanje zvuka. DSP je prisutan u svakom koraku moderne audio produkcije, od snimanja i obrade signala, do masteringa finalnog zvuka, čime se postiže optimalna kvaliteta zvuka za različite medije [14].

3.5.2. Prednosti DSP-a u real-time aplikacijama

Jedna od ključnih prednosti DSP-a je sposobnost obrade signala u stvarnom vremenu, što je posebno važno u aplikacijama kao što su telekomunikacije, multimedija i kontrolni sustavi. U telekomunikacijama, DSP procesori omogućuju brzo filtriranje i obradu dolaznih signala, kao što su glasovni i video pozivi, osiguravajući visoku kvalitetu usluge bez kašnjenja. Ova tehnologija omogućuje učinkovitu obradu signala u realnom vremenu, smanjujući odjek i prilagođavajući glasnoću na optimalnu razinu tijekom trajanja poziva [13].

U automobilskim sustavima, DSP procesori su ključni za napredne audio sustave koji uključuju surround zvuk i filtriranje pozadinskih šumova. DSP omogućuje obradu više kanala zvuka u stvarnom vremenu, pružajući visokokvalitetno audio iskustvo čak i u složenim okruženjima, kao što su automobili. Ovi procesori također omogućuju dinamičku prilagodbu zvuka ovisno o akustičnim uvjetima u vozilu, što rezultira jasnijim i uravnoteženijim zvukom [14].

Još jedna važna primjena DSP-a je u kontrolnim sustavima i obradi signala iz senzora. U industrijskim primjenama, DSP omogućuje brzu analizu podataka sa senzora u realnom vremenu, što je ključno za sustave automatizacije i nadzora. Na primjer, u robotskim sustavima, DSP procesori obrađuju informacije iz više senzora istovremeno, omogućujući robotičkim sustavima brze reakcije na promjene u okolini [13].

3.6. Aplikativno specifičan integrirani sklop (ASIC)

3.6.1. Dizajn i prilagodba za specifične zadatke

Aplikativno specifično integrirani sklopovi (ASIC) predstavljaju visoko optimizirane čipove dizajnirane za izvršavanje točno definiranih zadataka uz maksimalnu učinkovitost. Za razliku od općenitih procesora, kao što su CPU-ovi i GPU-ovi, ASIC-ovi se projektiraju prema vrlo specifičnim zahtjevima aplikacija, što omogućuje optimizaciju svakog aspekta čipa, od brzine obrade do potrošnje energije. Ovaj pristup prilagodbe rezultira visokim performansama jer se svaki dio ASIC-a dizajnira kako bi maksimalno iskoristio resurse potrebne za određeni zadatak. Na primjer, ASIC-ovi su široko korišteni u primjenama gdje je potreban visok nivo obrade podataka, poput kriptovaluta (rudarenje bitcoina) ili naprednih sustava za strojno učenje [15][16].

ASIC-ovi su posebno korisni u području umjetne inteligencije, gdje omogućuju ubrzavanje izvođenja algoritama dubokog učenja. Usporedno s čipovima za generalnu upotrebu, poput GPU-ova, ASIC-ovi su optimizirani za specifične funkcije kao što su matriksne multiplikacije, koje se često koriste u treniranju neuronskih mreža. Na primjer, Tensor Processing Units (TPU), koje je razvila kompanija Google, predstavljaju tip ASIC-a dizajniranog posebno za ubrzanje modela dubokog učenja [16].

Uz specijaliziranu arhitekturu, ASIC-ovi također su razvijeni za uređaje poput pametnih telefona i specijaliziranih kamera. U pametnim telefonima, ASIC-ovi se često koriste za zadatke poput prepoznavanja lica, gdje prilagodljiva obrada slike omogućuje brže i preciznije rezultate u realnom vremenu. U specijaliziranim kamerama, ASIC-ovi optimiziraju algoritme za obradu slike, čime se poboljšava kvaliteta slike i smanjuje vrijeme potrebno za analizu podataka [15].

ASIC-ovi su također ključni u sustavima visokofrekventnog trgovanja (HFT). Zbog svoje sposobnosti da izvršavaju specifične zadatke s ultra-niskom latencijom, ASIC-ovi omogućuju financijskim institucijama brzu obradu podataka i donošenje odluka u realnom

vremenu. Zahvaljujući specifičnoj optimizaciji za ove zadatke, ASIC-ovi pružaju vrhunsku brzinu obrade uz minimalnu potrošnju energije [15][16].

Jedan od glavnih izazova u razvoju ASIC-ova leži u njihovoj specijaliziranoj prirodi. Za razliku od CPU-ova i GPU-ova, koji su fleksibilni i mogu izvršavati širok raspon zadataka, ASIC-ovi često su dizajnirani za jednu određenu primjenu, što ih čini manje prilagodljivima u slučaju promjene zahtjeva ili novih tehnologija. No, zahvaljujući sve većoj potražnji za visoko specijaliziranim hardverom, ASIC-ovi se sve više koriste u industrijama koje zahtijevaju visoke performanse i nisku potrošnju energije [15][16].

3.6.2. Energetska učinkovitost i optimizacija

ASIC-ovi su poznati po svojoj energetskoj učinkovitosti, jer su dizajnirani da izvršavaju samo specifične zadatke za koje su razvijeni, eliminirajući potrebu za viškom resursa ili nepotrebnim funkcijama. Usporedno s univerzalnim procesorima, poput CPU-ova i GPU-ova, koji su dizajnirani za širok raspon zadataka, ASIC-ovi se prilagođavaju isključivo zahtjevima specifične aplikacije, što rezultira značajnim smanjenjem potrošnje energije. Ova optimizacija omogućuje ASIC-ovima da budu izuzetno efikasni u aplikacijama gdje se zahtijeva intenzivna obrada podataka, poput dubokog učenja, autonomnih vozila i aplikacija u oblaku [15][16].

Jedan od ključnih aspekata dizajna ASIC-ova je primjena tehnike dinamičkog skaliranja napona i frekvencije (DVFS), koja omogućuje procesoru prilagodbu svoje snage i frekvencije ovisno o trenutnim radnim zahtjevima. Kada procesor nije u potpunosti opterećen, smanjuje napon i frekvenciju kako bi uštedio energiju. Na taj način, ASIC-ovi uspijevaju održati visoke performanse uz minimalnu potrošnju energije, što je ključno za dugotrajne radne zadatke ili aplikacije koje zahtijevaju kontinuiranu obradu [15].

ASIC-ovi koriste napredne metode termičkog upravljanja za smanjenje pregrijavanja tijekom intenzivnih operacija. Zahvaljujući preciznim tehnikama hlađenja i optimizacije distribucije topline, ASIC-ovi mogu raditi na maksimalnim performansama bez rizika od pregrijavanja, što dodatno smanjuje potrebu za vanjskim sustavima hlađenja i čini ih efikasnijima u odnosu na univerzalne procesore [16].

Primjerice, u algoritmima dubokog učenja, ASIC-ovi omogućuju prilagodbu takta i napajanja prema potrebama aplikacija, osiguravajući stabilnu radnu temperaturu i dugotrajan rad [15].

Pored smanjenja potrošnje energije i termičke optimizacije, AI algoritmi se sve češće koriste za optimizaciju dizajna ASIC-ova. Umjetna inteligencija pomaže inženjerima u analizi performansi čipova i njihovih energetskih zahtjeva, predlažući prilagodbe koje omogućuju

bolje upravljanje resursima bez žrtvovanja performansi. Ova vrsta optimizacije igra ključnu ulogu u razvoju novih ASIC-ova, osobito u kontekstu visokih zahtjeva za računalnom snagom u područjima poput oblačnog računalstva i autonomnih sustava [16].

ASIC-ovi su također važni u oblaku i podatkovnim centrima, gdje njihova energetska učinkovitost pomaže u smanjenju ukupnih troškova rada i održavanja. Na primjer, u velikim podatkovnim centrima, gdje se zahtijeva kontinuirano izvođenje složenih operacija, ASIC-ovi optimiziraju energetske resurse, što omogućuje značajno smanjenje potrošnje energije na globalnoj razini. Uz to, ovi čipovi smanjuju potrebu za složenim sustavima hlađenja, što dodatno doprinosi održivosti i smanjenju troškova u dugoročnom radu [15][16].

Jedna od najvažnijih primjena ASIC-ova je u autonomnim vozilima, gdje je potrebna brza i pouzdana obrada podataka s minimalnim kašnjenjem. Zbog svoje specifične prilagodljivosti, ASIC-ovi omogućuju vozilima da obrađuju informacije sa senzora u realnom vremenu, osiguravajući sigurnost i učinkovitu vožnju. Ova tehnologija koristi napredne algoritme dubokog učenja za prepoznavanje objekata, donošenje odluka i upravljanje vozilom u složenim prometnim situacijama, sve uz minimalnu potrošnju energije [15].

U kontekstu oblačnog računalstva, ASIC-ovi se koriste za optimizaciju rada servera i podatkovnih centara. Zbog svoje energetske učinkovitosti i optimiziranih performansi, omogućuju učinkovitu obradu podataka u velikim razmjerima, smanjujući potrebu za velikim brojem univerzalnih procesora i dodatnim resursima za hlađenje. Na taj način, ASIC-ovi doprinose smanjenju ugljičnog otiska podatkovnih centara, čineći ih održivijima i ekološki prihvatljivima [16].

4. Učinak vertikalnog skaliranja na specifične zadatke (Praktični dio)

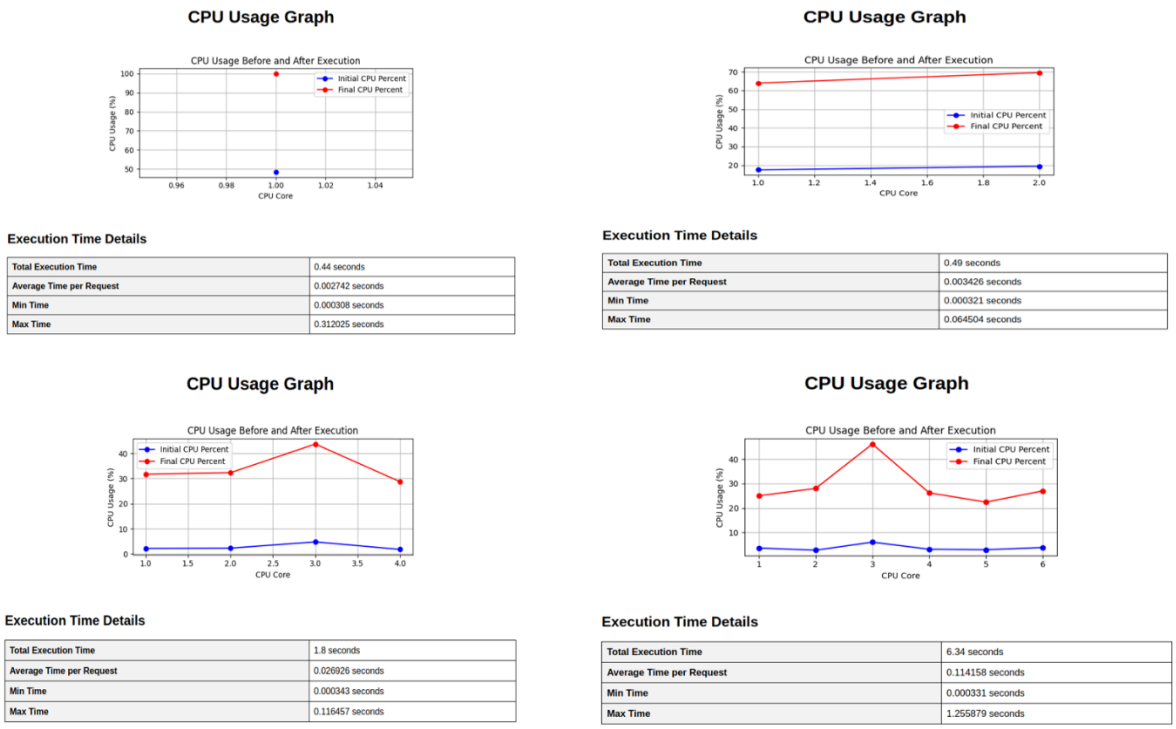
4.1. Opis eksperimenta

Projekt je dostupan na poveznici: https://github.com/dujebuljat/Zavrsni-demo

Eksperiment koji je proveden unutar Django aplikacije fokusirao se na ispitivanje utjecaja broja dostupnih procesorskih jezgri na performanse specifičnih zadataka, poput enkripcije i dekripcije stringova. U sklopu eksperimenta, program generira nasumičan string duljine 5000 znakova koji se zatim enkriptira i dekriptira kroz 500 iteracija. Primarni cilj eksperimenta bio je promatrati promjene u CPU opterećenju (CPU load) prije i nakon izvršenja ovih operacija, pri čemu su testirane različite konfiguracije s 1 do 6 dostupnih jezgri.

Aplikacija je koristila multiprocesiranje kako bi se omogućilo paralelno izvršavanje zadataka, s očekivanjem da bi povećanje broja jezgri trebalo smanjiti ukupno vrijeme izvršenja. Međutim, kako se teoretski raspravlja u prethodnim dijelovima rada, povećanje broja jezgri donosi izazove poput sinkronizacije i pristupa memoriji, što može ograničiti linearni rast performansi. Ovaj eksperiment je također istraživao utjecaj cache koherencije i memorijskih uskih grla u višejezgrenim sustavima.

4.2. Prikaz rezultata



Slika 7. Grafovi opterećenja procesora: 1, 2, 4, 6 jezgri

Rezultati eksperimenta prikazani su putem nekoliko ključnih grafova koji ilustriraju kako se opterećenje procesora mijenja prije i nakon izvršenja programa na različitim konfiguracijama procesora, konkretno s 1, 2, 4 i 6 jezgri, kao što je prikazano na spojenim slikama. Ovi grafovi jasno pokazuju trendove vezane uz opterećenje CPU-a i vrijeme izvršenja zadataka, a temeljna svrha je analizirati učinkovitost vertikalnog skaliranja.

Grafovi opterećenja procesora

Grafovi prikazuju opterećenje procesora prije i nakon izvršenja programa. U testiranjima s jednom i dvije jezgre, CPU opterećenje je značajno veće tijekom izvršenja programa, što ukazuje na to da ove konfiguracije rade blizu maksimalnog kapaciteta, s vrlo visokim konačnim opterećenjem procesora. Kada se broj jezgri poveća na četiri i šest, opterećenje se bolje raspoređuje među jezgrama, ali ne dolazi do proporcionalnog smanjenja opterećenja na svakoj pojedinoj jezgri, što potvrđuje prisutnost uska grla memorije i problema sinkronizacije između jezgri.

Kod konfiguracije s četiri jezgre, vidljivo je blago smanjenje opterećenja, ali završno opterećenje ostaje relativno visoko na trećoj jezgri, što pokazuje da sve jezgre nisu ravnomjerno opterećene, sugerirajući neoptimalnu distribuciju zadataka. Sa šest jezgri, opterećenje se još više raspodjeljuje, ali rezultati i dalje pokazuju određeno zasićenje performansi — dodatne jezgre ne rezultiraju značajnim povećanjem ukupnih performansi, a visoko opterećenje na nekim jezgrama ukazuje na memorijska ograničenja i potrebu za boljom sinkronizacijom.

Analiza grafova opterećenja procesora

Grafovi opterećenja procesora jasno ilustriraju nekoliko ključnih trendova. Kod manje jezgrenih sustava, poput testova s jednom ili dvije jezgre, dolazi do vrlo visokog finalnog opterećenja, što upućuje na gotovo potpuno iskorištavanje procesorskih resursa. U tim slučajevima, CPU radi s maksimalnim opterećenjem tijekom cijelog procesa, s malim razmacima između inicijalnog i finalnog opterećenja.

Povećanjem broja jezgri na četiri i šest, vidimo ravnomjerniju raspodjelu opterećenja između jezgri. Međutim, dolazi do stagnacije u smanjenju opterećenja, što upućuje na ograničenja u paralelizaciji zadataka i pojačava prethodne teze o problemima cache koherencije i zajedničkog pristupa memoriji. Kod konfiguracija s višim brojem jezgri, pojedine jezgre pokazuju nisko početno opterećenje, ali završno opterećenje značajno raste, ukazujući na to da procesor mora koordinirati pristup memoriji između više jezgri, što rezultira latencijama i neiskorištenim potencijalom dodatnih jezgri.

Vremenske vrijednosti i analiza

Tablice s vremenskim rezultatima, koje uključuju ukupno vrijeme izvršenja, prosječno vrijeme po zahtjevu, minimalno i maksimalno vrijeme, pružaju dodatne uvide u performanse sustava. U testiranjima s jednom jezgrom, ukupno vrijeme izvršenja je 0.44 sekunde, dok kod testiranja sa šest jezgri vrijeme raste na 6.34 sekunde. Ovaj porast ukupnog vremena pokazuje da povećanje broja jezgri ne dovodi nužno do smanjenja vremena izvršenja, već može dovesti do zasićenja sustava, gdje višak jezgri ne rezultira proporcionalnim smanjenjem vremena zbog problema sinkronizacije.

Prosječno vrijeme po zahtjevu također pokazuje zanimljive rezultate. Kod jedne jezgre, prosječno vrijeme po zahtjevu je 0.0027 sekundi, dok kod šest jezgri iznosi 0.1141 sekundu, što ukazuje na to da, iako su pojedinačne jezgre manje opterećene, povećanje broja jezgri dovodi do neefikasnosti u raspodjeli zadataka, vjerojatno zbog problema povezanih s koherencijom cache-a i zajedničkim pristupom memoriji.

Minimalno i maksimalno vrijeme po zahtjevu dodatno potvrđuju ove nalaze. Dok se minimalna vremena smanjuju s povećanjem broja jezgri, maksimalno vrijeme po zahtjevu raste kako broj jezgri raste, što ukazuje na kašnjenja uzrokovana sinkronizacijom i pristupom memoriji između jezgri. Ovi podaci jasno pokazuju da linearno povećanje broja jezgri ne donosi linearno smanjenje vremena izvršenja, već rezultira smanjenjem efikasnosti zbog hardverskih ograničenja.

4.3. Zaključak o utjecaju vertikalnog skaliranja

Rezultati eksperimenta potvrđuju teorijske tvrdnje o ograničenjima vertikalnog skaliranja. Dok povećanje broja jezgri omogućuje bolju raspodjelu opterećenja i smanjenje ukupnog vremena izvršenja zadatka, zasićenje performansi postaje očito nakon određene točke. Ovaj fenomen proizlazi iz ograničenja arhitekture višejezgrenih procesora, osobito u kontekstu koherencije cache-a i memorijskih uskih grla, gdje više jezgri koje istovremeno pristupaju zajedničkim resursima izazivaju kašnjenja.

Zaključujemo da, iako vertikalno skaliranje pruža određene prednosti u smislu paralelizacije, povećanje broja jezgri ne rezultira uvijek linearnim povećanjem performansi. Praktični rezultati jasno pokazuju da stvarne performanse višejezgrenih sustava ovise o arhitekturalnim ograničenjima i memorijskim uskim grlima, što potvrđuje tezu da samo povećanje broja jezgri nije dovoljno za postizanje drastičnog poboljšanja ukupnih performansi. Ova saznanja upućuju na potrebu za daljnjim istraživanjima i razvojem novih arhitektura koje će prevladati trenutna ograničenja višejezgrenih procesora i omogućiti bolje iskorištavanje dostupnih resursa.

Literatura

[1] D. Etiemble, »45-year CPU evolution: one law and two equations,« arXiv.org, 1. ožujak 2018. [Mrežno]. Dostupno: https://arxiv.org/abs/1803.00254. [Pristupljeno 15. kolovoz 2024.].

[2] X. Jiang, »CMOS technology scaling and its implications,« u *Digitally-Assisted Analog and Analog-Assisted Digital IC Design,* Cambridge University Press, 2015., pp. 1-10.

[3] G. Arcuri i S. Shivakumar, »Moore's Law and Its Practical Implications,« Center for Strategic & International Studies, 18. listopad 2022. [Mrežno]. Dostupno: https://www.csis.org/analysis/moores-law-and-its-practical-implications. [Pristupljeno 15. kolovoz 2024.].

[4] A. Tiwari, »Performance Comparison of Cache Coherence Protocol on Multi-Core Architecture,« svibanj 2014. [Mrežno]. Dostupno: http://ethesis.nitrkl.ac.in/6165/. [Pristupljeno 16. kolovoz 2024.].

[5] S. Deshpande, P. Ravale i S. Apte, »Cache coherence in centralized shared memory and distributed shared memory architectures,« 2010. [Mrežno]. Dostupno: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=96a4a78397a581e27d989e8c42da490051218042. [Pristupljeno 16. kolovoz 2024.].

[6] W. Luo, R. Fan, Z. Li, D. Du, Q. Wang i X. Chu, »Benchmarking and Dissecting the Nvidia Hopper GPU Architecture,« 21. veljača 2024. [Mrežno]. Dostupno: https://ar5iv.labs.arxiv.org/html/2402.13499. [Pristupljeno 18. kolovoz 2024.].

[7] Y. Xue, Y. Liu, L. Nai i J. Huang, »Hardware-Assisted Virtualization of Neural Processing Units for Cloud Platforms,« 7. kolovoz 2024. [Mrežno]. Dostupno: https://ar5iv.labs.arxiv.org/html/2408.04104. [Pristupljeno 19. kolovoz 2024.].

[8] Intel, »Quick overview of Intel's Neural Processing Unit (NPU),« Intel Corporation, [Mrežno]. Dostupno: https://intel.github.io/intel-npu-acceleration-library/npu.html. [Pristupljeno 20. kolovoz 2024.].

[9] D. Sanmartin i V. Prohaska, »Exploring TPUs for AI applications,« 16. rujan 2023. [Mrežno]. Dostupno: https://arxiv.org/pdf/2309.08918. [Pristupljeno 20. kolovoz 2024.].

[10] Fidus, »FPGA and Machine Learning: Unlocking the Future of AI Hardware,« Fidus, 9. rujan 2024. [Mrežno]. Dostupno: https://fidus.com/blog/fpga-and-machine-learning-unlocking-the-future-of-ai-hardware/. [Pristupljeno 24. kolovoz 2024.].

[11] P. K. Seng, J. P. Lee i M. L. Ang, »Embedded Intelligence on FPGA: Survey, Applications and Challenges,« 8. travanj 2021. [Mrežno]. Dostupno: https://www.mdpi.com/2079-9292/10/8/895. [Pristupljeno 24. kolovoz 2024.].

[12] A. Boutros, E. Nurvitadhi, R. Ma, S. Gribok, Z. Zhao, J. C. Hoe, V. Betz i M. Langhammer, »Beyond Peak Performance: Comparing the Real Performance of AI-Optimized FPGAs and GPUs,« 2020. [Mrežno]. Dostupno: https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/a1153843-beyond-peak-performance-white-paper.pdf. [Pristupljeno 26. kolovoz 2024.].

[13] C. M. Jagielski, »Design of a real-time digital signal processing audio processing technique,« svibanj 2012. [Mrežno]. Dostupno: https://oaktrust.library.tamu.edu/server/api/core/bitstreams/b26768cd-d842-474d-b354-251c675d7ef2/content. [Pristupljeno 27. kolovoz 2024.].

[14] C. Golembeski, »Digital Signal Processing (DSP) Audio Explained: A Beginner's Guide,« Headphonesty, 10. listopad 2023. [Mrežno]. Dostupno: https://www.headphonesty.com/2022/01/dsp-audio/. [Pristupljeno 27. kolovoz 2024.].

[15] AnySilicon, »Artificial Intelligence (AI) in ASIC/SoC Design Today and Future,« AnySilicon, 2021. [Mrežno]. Dostupno: https://anysilicon.com/artificial-intelligence-ai-in-asic-soc-design-today-and-future/. [Pristupljeno 2. rujan 2024.].

[16] D. Harris, »Sustainable Strides: How AI and Accelerated Computing Are DIviding Energy Efficiency,« NVIDIA, 22. srpanj 2024. [Mrežno]. Dostupno:

https://blogs.nvidia.com/blog/accelerated-ai-energy-efficiency/. [Pristupljeno 4. rujan 2024.].

[17] I. A. Amory, A. H. Ahmed i L. F. Jumma, »MESI protocol for multicore processors based on FPGA,« siječanj 2021. [Mrežno]. Dostupno: http://pen.ius.edu.ba/index.php/pen/article/view/1772. [Pristupljeno 6. rujan 2024.].

Summary

Vertical Scaling Techniques for Computer Performance Enhancement

This paper focuses on the performance analysis of multicore processors, with a particular emphasis on the challenges of vertical scaling, such as memory bottlenecks and cache coherence. Benchmark tests conducted in a virtual environment show that increasing the number of processor cores can lead to performance saturation due to concurrent access to the same memory space. In the practical part of the paper, a web application was developed to test encryption and decryption using multicore processors and to measure CPU load before and after task execution, highlighting scalability and efficiency challenges.

Additionally, the paper explores specialized processors such as graphics processing units (GPUs), neural network processors (NPUs), and tensor processing units (TPUs). These processors leverage specific architectures that overcome the limitations of traditional CPU systems, enabling faster and more efficient data processing in applications such as artificial intelligence, deep learning, and massive parallelization. The research results confirm the theoretical assumptions about the limitations of traditional vertical scaling and indicate the need for applying alternative architectures to achieve optimal computational efficiency.

Keywords: multicore processors, cache coherence, vertical scaling, processor performance, specialized processors

Popis slika