Advanced research methods for multi-objective optimization of CPUs architectures

Metode avansate de cercetare pentru optimizarea multi-obiectiv a arhitecturilor de procesare

Teză de abilitare
Rezumat

Adrian FLOREA

2021

Această teză de abilitare reprezintă un sumar al celor mai importante realizări personale în domeniul arhitecturilor de procesoare, cu accent pe perioada ulterioară susţinerii tezei de doctorat. Evident că, cu mici excepţii, la munca din spatele obţinerii acestor realizări am avut cel puţin un colaborator, fie pe mentorul ştiinţific – dl. profesor Vinţan Lucian, fie pe alţi colegi din universitate sau din străinătate (co-autori la lucrările ştiinţifice publicate). Astfel, este oarecum nedrept că există un singur autor al tezei de abilitare deoarece datorez recunoştinţă colaboratorilor mei pentru tot ce am realizat. Pe de altă parte, cel puţin în domeniul microarhitecturilor, sunt foarte puţine cazuri în care activitatea de cercetare este depusă de o singură persoană, de cele mai multe ori sunt colective de 3-4 cercetători care lucrează împreună pe anumite tematici. În afara acestor contribuții, intenția este de a ilustra viziunea mea asupra domeniului arhitecturilor de procesoare în care sunt plasate contribuțiile mele anterioare. Având în vedere că cercetările mele în acest vast şi de actualitate domeniu datează de mai bine de douăzecişidoi de ani, această sinteză a activităţii din ultimii şaisprezece ani se întinde pe un spectru destul de larg ţinând cont şi de dinamicitatea domeniului, rezultatele fiind inevitabil legate de efortul depus în întreaga activitate de cercetare. Pe lângă caracterul științific, lucrarea are un puternic aspect tehnic datorită multor simulări dezvoltate. Autorul sintetizează metodologiile de simulare utilizate în cercetare, instrumentele software de compilare, depanare și simulare, benchmark-urile standardizate SPEC (*Standard Performance Evaluation Corporation*) pentru arhitecturile monocore și SPLASH (*Stanford ParalleL Applications for Shared memory*) pentru arhitecturile multicore. Deşi valoarea adăugată este mai degrabă de natură tehnică şi nu ştiinţifică, utilitatea sa în cadrul tezei de abilitare este indiscutabilă.

Primul capitol cuprinde o motivaţie a implicării mele în acest domeniu, o introducere în problematica microarhitecturilor provocările și limitările existente, precum şi calea pe care am urmat-o în a sintetiza prezenta lucrare. Totodată, am încercat să punctez elementele importante ale activităţii mele de cercetare şi să rezum modestele mele contribuţii la dezvoltarea domeniului. Impactul cercetărilor a fost reliefat în indici scientometrici şi în prestigiul publicaţiilor ce citează lucrările publicate. Un lucru demn de subliniat este faptul că peste 42% din articolele publicate au ca şi co-autor un student la una din formele de învăţământ (licență, master sau doctorat). Colaborarea a fost extrem de utilă pentru ambele părți profesor / student în vederea inițierii studenților în cercetare și creării unei structuri în permanentă dezvoltare la nivelul centrului de cercetare Advanced Computer Architecture and Processing Systems (ACAPS) (http://acaps.ulbsibiu.ro/index.php) prin care să aibă loc transferul de cunoștințe de la profesor la doctoranzi, de la doctoranzi la masteranzi și de la masteranzi la studenți, fiecare dintre aceștia parcurgând treptele dezvoltării profesionale și învățând de la înaintașii (mentorii) lor. Personal am muncit alături de ei şi am avut împreună satisfacţia muncii împlinite.

Capitolul al doilea prezintă limitarea *Memory Wall* cauzată de creșterea decalajului dintre viteza procesorului și lentoarea memoriei principale (DRAM) și necesitatea folosirii unui sistem ierarhic bazat pe unul sau mai multe niveluri de memorii cache, separate sau unificate. Contribuțiile personale au fost în ton cu dezvoltările tehnologice ale vremurilor și provocările curente ale domeniului, de la concepte precum (Selective) Victime Cache (propuse inițial de Norman Jouppi [1], continuate de Stiliadis [2], implementate în procesoarele comerciale AMD [3] și folosit ca și concept fundamental în alte investigații de tipul *cache dead blocks* [4], *victim buffer in embedded*

system design [5], etc) la influența negativă în procesoare cu mii de instrucțiuni *in-flight* produsă de aplicațiile de tip *pointer chasing* în care are loc serializarea execuției operațiunilor cu referire la memorie caracterizate de latență ridicată. Abordările recente specifice memoriilor cache vizează cercetarea acestora nu doar din perspectivă tehnică ci și impactul la nivel societal actual, în principal, cerința de a reduce consumul de energie și de a reduce amprenta de carbon a algoritmilor de înlocuire a blocurilor din cache precum și asigurarea securității datelor memorate.

O provocare continuă în ştiinţa şi ingineria calculatoarelor o reprezintă înţelegerea profundă a complexelor şi subtilelor interacţiuni hardware-software. Optimizările interfeţei hardware-software nu sunt posibile decât pe baza unui proces de înţelegere, care necesită o viziune integratoare asupra multiplelor şi complicatelor procese de prelucrare a informaţiei, codificată la niveluri semantice diferite. În acest sens, în capitolul al treilea sunt prezentate cercetările autorului relative la analiza legăturilor adânci dintre caracteristicile unor programe procedurale și obiectuale HLL (High Level Languages), pe de o parte, şi execuția lor pe arhitecturile cu paralelism la nivelul instrucțiunilor, respectiv structurile hardware de predicţie a salturilor indirecte prin registru, pe de altă parte. O parte din contribuțiile ilustrate în acest capitol au fost incluse în teza de doctorat, etapă fundamentală în cariera mea profesională, care a culminat cu formarea mea ca cercetător. Aportul adus relativ recent referitor la influența compilatorului asupra arhitecturilor de procesare s-a concretizat prin implementarea unor metode de optimizare bazate pe algoritmi genetici și de inteligență colectivă (*swarm intelligence*) pe problema de colorare a grafurilor aplicată pentru alocarea regiștrilor de către compilator în sistemele încorporate.

În capitolul 4 este prezentată limitarea fundamentală a paradigmei *ILP Wall* intitulată *fetch bottleneck / control flow bottleneck* (limitarea producătorului) și metodele predictive de creștere a performanței aplicate fluxului de control al programului. Sunt ilustrate tehnici de predicție a salturilor condiționate de la structuri adaptive corelate pe două niveluri la predictoare markoviene și neuronale precum și soluții de precalculare a adresei de salt. Tehnici aparte de predicție sunt aplicate categoriei de salturi și apeluri indirecte, cele mai de succes fiind predictoarele hibride, cascadate sau metapredictoarele. Calitatea unui model de predicție depinde însă în mare măsură de calitatea datelor disponibile dar și de alegerea caracteristicilor schemei de predicție. Marea majoritate a schemelor se bazează pe utilizarea unui număr mai mare de caracteristici de intrare (cum ar fi adresa instrucțiunii de salt, comportamentul anterior din perspectivă globală a tuturor salturilor din program sau locală strict din perspectiva saltului supus predicției etc.) fără a lua în considerare cauza reală – salturile nepolarizate care sunt dificil de prezis, apar cu un comportament aproape aleator în același context de intrare, au o acuratețe de predicție scăzută și implicit generează pierderi de performanță. Bazat pe o abordare hibridă dintre Matematică și Calculatoare, am dezvoltat patru metrici referitoare la gradul de aleatorism al instrucţiunilor de salt nepolarizate: complexitatea Kolmogorov a programelor de calculator, rata de compresie, entropia discretă și acurateţea predicției folosind predictoare Markov cu straturi ascunse (Hidden Markov Models). Toate aceste metrici pot ajuta proiectantul de sisteme de calcul să înțeleagă mai bine predictibilitatea ramificaţiilor de program şi dacă predictorul poate fi îmbunătăţit astfel încât să prezică cu o mai mare precizie salturile nepolarizate.

Capitolul al cincilea prezintă limitarea căii critice de program intitulată *issue bottleneck / data flow bottleneck* (limitarea consumatorului) împreună cu două tehnici anticipative şi speculative propuse pentru depăşirea acestei limitări şi anume, reutilizarea dinamică a instrucţiunilor şi respectiv

predicţia dinamică a valorilor instrucţiunilor. Ambele soluţii se bazează pe principiul vecinătăţii temporale a valorilor instrucţiunilor (*value locality*), formulat în 1996 de patru colective internaţionale de cercetare de la AMD Nexgen, Carnegie Mellon, Wisconsin (USA) şi Technion (Israel). Autorul acestei teze extinde conceptul asupra regiştrilor procesorului MIPS cu avantaje evidente, în primul rând reducerea drastică a numărului de predictoare implementate, dezvoltarea de predictoare hibride de valori (contextuale şi incrementale) în care selecţia predictorului curent se face prin intermediul unor selectoare dinamice de tip automat finit respectiv de tip neuronal, precum și ilustrarea unor consideraţii interesante asupra performanţei globale de procesare. O idee de cercetare originală a constat în implementarea în mod selectiv a tehnicilor anticipative şi speculative doar pentru valorile produse de instrucțiunile cu latență ridicată (Înmulțire / Împărțire sau instrucțiuni Load critice – care accesează cu miss primul nivel din cache și ajung în fruntea buffer-ului de reordonare), prezentându-se totodată și impactul la nivel de performanță de procesare și energie consumată.

În capitolul 6 sunt explorate oportunități de cercetare în arhitecturile multicore apărute ca o soluție la limitarea de tip *Power Wall* arătându-se că fertilizarea încrucișată între domeniul arhitecturilor de procesoare, calculul evoluționist cu metode de optimizare multi-obiectiv și folosirea regulior fuzzy pentru reprezentarea cunoștințelor duce la un design eficient al procesorului. Obiectivul principal îl reprezintă optimizarea din perspectivă multiplă a arhitecturilor de procesare considerând ca metrici performanța, energia consumată, aria de integrare și temperatura disipată. Se motivează necesitatea implementării unor instrumente de explorare automată a spațiului de proiectare (*automatic design space exploration*) aferent procesoarelor mono/multicore cu peste 20 de parametri arhitecturali bazate pe algoritmi genetici de tip *Pareto* (*Non-dominated Sorting Genetic Algorithm* versiunea a 2-a), care după rularea unui număr consistent de generații permite analiza comparativă a rezultatelor folosind metrici de tipul: *front Pareto*, *hypervolum*, *coverage*, etc. O abordare originală o reprezintă îmbunătățirea algoritmului de explorare automată a spațiului de proiectare folosind cunoștințe de domeniu implementate sub forma unor reguli *fuzzy* în operatorii genetici care au sporit diversitatea și calitatea soluțiilor precum și viteza de convergență a procesului de explorare. Provocările actuale urmăresc îmbunătățirea procesului de explorare prin folosirea de tehnici de tip *Response Surface Modeling* (RSM) și identificarea modului de reprezentare adecvată a cunoștințelor de domeniu în sistemele multicore prin răspunsul la întrebările: care sunt regulile (mai importante), sunt acestea contradictorii, cum pot fi implementate în operatorii genetici de selecție, mutație sau încrucișare?

Capitolul 7 sintetizează planurile de viitor pe atât din perspectivă didactică, educaţională cât şi din perspectiva cercetării ştiinţifice, a creşterii capitalului uman din universitate şi poate chiar cea a transferului tehnologic spre companii.